ON THE ORDERING OF THE $t = 2$ BEST OF n ITEMS USING BINARY COMPARISONS

by

Milton Sobel[§]
University of Minnesota
Imperial College

Technical Report No. 113

[§]Supported partially by a National Institutes of Health Special Fellowship while at Imperial College, London, and partially by National Science Foundation Grant Number Gp-9018, through the University of Minnesota.

1. Introduction

The problem considered is that of ranking the t best (i.e., largest) of n unequal numbers (or objects with respect to an associated scalar such as weight) when only binary errorless comparisons are allowed. In some applications, these n numbers are unknown, but in others, e.g., the "sorting problem," the numbers are actually known. Here, a machine (or a person) starts with a sequence of n numbers in random order and uses only binary comparisons to put them all in (say) ascending order. In the application to aligning n tennis players according to ability, we call this a "tournament problem." We assume that the players have unequal ability (or skill), that the better player always wins, and that the relation "better than" is transitive. If we have n unequal weights and a simple balance that only allows one weight on each pan, then this problem (of ordering the n weights) is called a "weighing problem." From the point of view of questionnaire theory (which emphasizes the graph-theoretic and information-theoretic nature of the problem), this is called the problem of 'tri.' These are clearly all the same problem, corresponding to $t = n$ (or equivalently $t = n - 1$) in our formulation, and we prefer to call it the "Steinhaus expectation problem" for $t = n - 1$ because of the early interest Steinhaus showed in a related minimax problem (see below).

It is assumed that the n numbers are initially in random order, i.e., either their order has been randomized, or we are willing to assume this. To explain our goal, consider the number T of binary comparisons (or tests) required for $n = 3$. Already T is not constant ($T = 2$ or 3), and from the initial random order, we obtain the expectation $E\{T|n = 3\} = \frac{8}{3}$ for the optimal procedure. Our main goal is to find a procedure R that minimizes this expectation. Several new procedures are introduced in this paper, all with expectations below that of the Steinhaus procedure defined below. Some of these have values smaller than any procedure known to the author, and some are conjectured to be optimal.

Another goal of this paper is to find a procedure R that minimizes the maximum number of tests required to guarantee that we can order the t best of n numbers; we refer to this as the "Steinhaus minimax problem." The expectation and minimax goals are not unrelated, and for small values of n, we can find procedures both E-optimal (i.e., with smallest expectation) and M-optimal (i.e., with smallest maximum).

Steinhaus [23] gives a basic, fully inductive procedure R_S for the minimax goal. In the 1950 edition of this book, he conjectures that this procedure is optimal for all n, but this is deleted in a later edition, and in another book [24] on problems, a counterexample is explicitly worked out for $n = 5$. Although the procedure R_S is at the "bottom" of our list of procedures for $t = n - 1$ (it has the largest expectation and the largest maximum length among all the procedures in the Section 5 tables), it represents an important standard for comparison partly because it is both E-optimal and M-optimal among the fully inductive procedures [10] and partly because more is known about its properties. Kislitsyn found general bounds for the expectation under R_S in [14] and derived an asymptotic expression for the same expectation in [15]. Although this procedure R_S is widely known in computer science (it is called *binary insertion* or *TID* or *ranking by insertion* or *binary search* by different authors), it is remarkable how many writers in this field assume either explicitly as on page 236 of [13] or implicitly that R_S is either E-optimal or M-optimal (or both) and are not familiar with other work in this area.

Another important procedure for both the M-goal and the E-goal is the semi-inductive procedure R_F of Ford and Johnson [8], although the paper is only concerned with the minimax problem. In fact, the procedure R_F is E-optimal for $n \leq 5$, and the expected values for moderate n (calculated by A. Hadian and the author) were found to be smaller than any others found in print at the start of

this investigation. Cesari [4] and Hadian [10] have modified the procedure R_F for $n \geq 6$ to obtain a smaller expectation without changing the M-value.

Picard [17] has given a procedure for $n = 6$ (and $t = 5$), which is both E-optimal and M-optimal. His approach through questionnaire theory combines a graph-theoretic and an information-theoretic analysis, which he applies to many interesting search problems.

For the sake of completeness, we should also mention the related papers of Bose, Nelson [1], and Hibbard [11] (see also the references in the latter), but because they apply restrictions on the number of locations in a computer that can be used or because their criterion is slightly different from our T or because their results are not in contention with ours, we omit their procedures in our comparisons. Also, our problem is related to merging ordered strings of numbers into a single string if the criterion is simply the number T of binary comparisons required and not the total number of key transfers as in Burge [2]. In the latter paper, it was empirically observed that our procedures were equally good under his (key-transfer) criterion but that his procedure was inferior under our T-criterion.

The main emphasis in this paper is on the use of 2 ideas for a testing procedure, namely pairing and expected uncertainty. Our entropy procedure R_E selects at each stage the comparison that maximizes the expected reduction in entropy due to a single comparison. Equivalently, it chooses the comparison that results in the smallest amount of uncertainty (or yields the maximum amount of information). By introducing certain types of pairing for the early comparisons, the procedure can be greatly simplified and, in some instances, actually improved. The idea of expected entropy was used for the group-testing problem by Sobel and Groll [22] and has also been used for other search problems by F. Dubail [7], who has called it "generalized entropy."

Our main interest is in one-step entropy procedures. A fairly obvious generalization of R_E, say $R_{E,g}$, which selects the comparison that maximizes the expected reduction in entropy in the next g tests ($g \geq 1$), can also be considered, as it is in [22] for the group-testing problem. All our procedures are such that they can make use of any *a priori* knowledge about the initial ordering as well as *a posteriori* knowledge gained at each stage.

The procedure $R_E = R_{E,1}$ (the pure one-step entropy procedure) gives optimal expectation results for small values of n ($n \leq 6$ for $t = 2$ and also for $t = n - 1$) wherever optimal procedures are known. In addition, each of the three entropy procedures consistently improves on known results for moderate values of n. In fact, it turns out to be interesting to find instances where R_E is not optimal. All our empirical results are consistent with a conjecture that an E-optimal procedure can be obtained from the procedure R_{E_1} or from the family $R_{E,g}$ with a moderately small value of g.

The case $t = 2$ will actually be treated first in this paper, before the case of $t = n - 1$, because it is a simpler problem and, at the same time, it exhibits the complexities associated with the case of general t ($1 \leq t \leq n - 1$).

The case of small t has a slight history of its own, starting with Lewis Carroll's essay [3] on the faulty manner (cup system) of awarding the second prize in a lawn tennis tournament in his day. He points out that if players are eliminated after one loss, then there is a high probability of not finding the correct second-best player. For example, with $n = 8$, under complete pairing (or so-called knock-out tournament that pairs off all the non-losers), the second-best player has a probability of $\frac{3}{7}$ of being in the same group of four as the best player and hence of not receiving the second prize.

The case $t = 2$ is discussed by Steinhaus [23], and the papers of J. Schreier [19] & J. Słupecki [20] are fundamental to our result that two of our procedures are M-optimal for $t = 2$. The case $t = 2$ has also been considered by Picard [17], and we use one of his procedures R_P in our table of

comparisons. For $t = 2$, we regard R_P as an analogue of the Steinhaus procedure R_S for $t = n - 1$, and we only consider procedures that are at least as good as R_P for the E-goal or the M-goal.

The work of David [5], Glenn [9], and Maurice [16] deals with knock-out, round-robin, and double-elimination tournaments and is related to our subject but not to the present paper. In their work, randomness is a result of associating more skills with a higher probability of winning. In our case, the better player always wins, and the randomness arises only from the initial random ordering of the n players. It is felt that a knowledge of the best procedures when there are distinct differences in skill (so that the better player always wins) should be helpful to design procedures for models that bring randomness into the observed results. A fine discussion of the work of David, Glenn, and Maurice on these types of tournaments is given in David [6].

Although no attempt is made in this paper to apply the techniques for large values of n or to find the procedure best for machine computation, the author feels that there is a challenge presented here to adapt the entropy procedure or some modification of it to large-scale machine computations for the large values of n. It is conjectured that the results will be substantially better than any others in print (see, e.g., Bose and Nelson [1]) even if one uses a slightly different criterion than the number of comparisons for comparing procedures.

2. Procedures for the Ordering Problem with $t = 2$

Several procedures are introduced, all of which are new, except for the procedure R_P due to Picard [17]. One of these procedures is an adaptation to $t = 2$ of the Ford-Johnson procedure and is denoted by R_F. One of the entropy procedures R_{E_1} is uniformly as good or better than any other procedure for all the values of n considered ($2 \leq n \leq 10$). Based on the work of Schreier [19] & Słupecki [20], two of the procedures are shown to be M-optimal. Each procedure is briefly described in this section, and a table of numerical comparisons is given; properties and derivations of results are given in Section 3.

We use the term 'fully inductive' to indicate a scheme in which the procedure for n players depends directly on that for $n-1$ players. The term 'semi-inductive' indicates that the scheme for n players depends directly on that for $\lfloor \frac{n}{2} \rfloor$ players, where $\lfloor x \rfloor$ is the largest integer $\leq x$. All logarithms in this paper are to the base two unless stated otherwise.

Procedure R_E: This is a one-step entropy procedure for $t = 2$ and is based on finding the binary comparison that maximizes the expected reduction in entropy after one comparison.

Procedure R_{E_1}: Suppose $n = 2^r + c$ $(0 \leq c < 2^r)$, and we conduct a knock-out tournament on the first (or any) 2^r players. Procedure R_{E_1} starts in this way and then uses the one-step entropy method to complete the problem.

For complete pairing and $n = 2^r + c$, we also want to allow pairing among the c remaining players; we then write $n = 2^{r_1} + 2^{r_2} + \ldots + 2^{r_s}$ $(r_1 > r_2 > \ldots > r_s \geq 0)$ and perform a knock-out tournament for each of these powers of two.

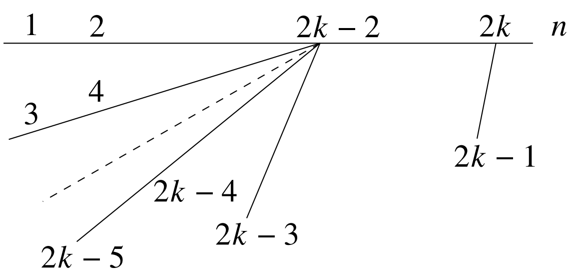
Procedure R_{E_2}: For this procedure, we do a complete pairing and then use the one-step entropy method to complete the problem.

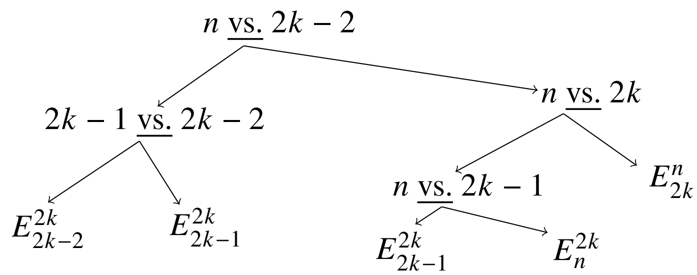
Procedure R_F: This is an analogue of the Ford-Johnson procedure applied to the case $t = 2$. Suppose $n = 2k$ or $2k + 1$. We describe the procedure in 3 steps.

1. Using ordinary pairing, we pair off $2k$ of the players for the first k comparisons, leaving one man out if n is odd.

2. Use induction (with the obvious procedures for $n = 2$ and 3) to order the $t = 2$ best among the k winners in step 1.

3. If n is even, step 2 results in an overall best player and two contenders for second best, thus requiring only one more comparison. If n is odd, we use a diagram for the third step. Let n or $2k + 1$ denote the player left out in steps 1 and 2, let $2k$ denote the winner in step 2, $2k - 1$ (resp., $2k - 2$) denote the contender that lost to $2k$ in step 1 (resp., step 2). The diagram and the continuation are given by





The left (resp. right) fork under a vs. b indicates that a loses to b (resp. a wins over b) and the endpoint E^a_b indicates the final result that a is best and b is second best.

Procedure R_{I^*}: This is a semi-inductive procedure without pairings. Let $n = 2k$ or $2k + 1$ as above. We first partition the n players into two subsets, each of size at least k, without making any comparisons, and then for $n \geq 4$, follow the three steps:

1. Use induction (with the obvious procedures for $n = 2$ and 3) to find the best player separately in each of the two subsets, keeping track of all contenders for second best.

2. Let the two winners play to determine the best player and put the loser (but not his inferiors) in contention for the second best. Suppose there are now $c \geq 2$ contenders for second best.

3. Use any simple knock-out tournament (with exactly $c - 1$ games) to determine the second-best.

Procedure R_M: For this procedure, we again use the binary expansion of n and complete pairing:

1. Find the best one separately in each of the subsets for which $r_i > 0$.

2. Play the best one of the smallest subset (of size 2^{r_s}) against the winner of the second smallest subset (of size $2^{r_{s-1}}$). Play this winner against the winner of the third smallest subset (of size $2^{r_{s-2}}$), etc., until the best one of all n is determined. Let c denote the number of contenders for second best.

3. Use any simple knock-out tournament with exactly $c - 1$ games to determine the second-best.

Procedure R_P: This fully inductive procedure for $t = 2$ due to Picard [17] is an analog of the Steinhaus procedure for $t = n - 1$. Let the players (in random order) be denoted by 1, 2, …, n; the iterative scheme of the procedure is described in 3 steps:

1. Play 1 vs. 2 and assume 1 loses to 2.
2. Play 3 vs. the loser 1. If 3 loses, then he is removed from contention. If 3 wins, then 1 is removed from contention, and 3 plays 2 to re-establish an ordering between the two top contenders.
3. Thus, in either case, we again have an ordered pair of contenders, and if there are new players left, we simply repeat the above scheme.

Although the procedure R_P is remarkable for its simplicity and amenability to analysis and machine computation, we later show that it is inadmissible. However, this procedure is useful as a standard for comparison for the E-problem and is conjectured to be optimal in the class of fully-inductive procedures for $t = 2$.

Procedure R_I: This is a semi-inductive procedure with the same first step as R_{I^*}, which we omit. We use the obvious procedures for $n = 2$ and 3 and assume that $n \geq 4$ in the following steps:

2. Use induction on each set separately to find both the best and the second-best players. Suppose $a \prec b$ and $c \prec d$ are the two pairs obtained, where $\prec$ denotes 'is inferior to.'
3. Play b vs. d and assume that d wins. Then, play b vs. c to determine the second best. Thus, for $n \geq 4$, step 3 consists of exactly two games.

Although R_I is quite poor in expectation, we include it for purposes of comparison and to illustrate the importance of subtle differences in procedure.

Comparison of Eight Procedures for the $t = 2$ Ordering Problem

Expected Values

Lower Bounds and Procedures	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$	$n=10$
LB[§]	1	2.584	3.917	4.922	5.773	6.488	8.380	9.057	9.668
CLB[#]	1	2.500	4.000	5.000	6.500	7.500	9.000	10.00	11.00
R_{E_1}	1	$2\frac{2}{3}$	4	$5\frac{8}{30}$	$6\frac{15}{30}$	$7\frac{170}{210}$	9	$10\frac{84}{1260}$	$11\frac{112}{3780}$
R_E	1	$2\frac{2}{3}$	4	$5\frac{8}{30}$	$6\frac{15}{30}$	$7\frac{170}{210}$	$9\frac{5}{840}$	N.C.	N.C.
R_{E_2}	1	$2\frac{2}{3}$	4	$5\frac{8}{30}$	$6\frac{20}{30}$	$7\frac{170}{210}$	9	$10\frac{84}{1260}$	$11\frac{112}{3780}$
R_F	1	$2\frac{2}{3}$	4	$5\frac{8}{30}$	$6\frac{20}{30}$	$7\frac{176}{210}$	9	$10\frac{160}{1260}$	$11\frac{1008}{3780}$
R_{I^*}	1	$2\frac{2}{3}$	4	$5\frac{12}{30}$	$6\frac{20}{30}$	$7\frac{180}{210}$	9	$10\frac{420}{1260}$	$11\frac{1512}{3780}$
R_M	1	$2\frac{2}{3}$	4	$5\frac{18}{30}$	$6\frac{20}{30}$	$7\frac{180}{210}$	9	$10\frac{840}{1260}$	$11\frac{2268}{3780}$
R_P	1	$2\frac{2}{3}$	$4\frac{1}{6}$	$5\frac{17}{30}$	$6\frac{27}{30}$	$8\frac{39}{210}$	$9\frac{366}{840}$	$10\frac{829}{1260}$	$11\frac{3243}{3780}$
R_I	1	$2\frac{2}{3}$	4	$5\frac{20}{30}$	$7\frac{10}{30}$	$8\frac{140}{210}$	10	$11\frac{840}{1260}$	$13\frac{1260}{3780}$

Minimax Values

	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$	$n=10$
LB[§§]	1	3	4	6	7	8	9	11	12
R_M	1	3	4	6	7	8	9	11	12
R_{I^*}	1	3	4	6	7	8	9	11	12
R_F	1	3	4	6	7	9	9	11	12
R_{E_2}	1	3	4	6	7	9	9	11	12
R_{E_1}	1	3	4	6	8	9	9	11	12
R_E	1	3	4	6	8	9	11	N.C.	N.C.
R_I	1	3	4	6	8	9	10	12	14
R_P	1	3	5	7	9	11	13	15	17

[§] This is a lower bound for all procedures using cycle pairing.

[#] CLB $= n - 2 + \frac{1}{2}\lfloor 2\log(n)\rfloor$ is a conjectured lower bound.

[§§] This M-lower bound due to Schreier is LB $= n - 1 + \lfloor \log(n-1)\rfloor$.

N.C. means not computed.

3. Formulas and Properties

Since our best results are for the simplest procedures, we consider our procedures in reverse order of their appearance in Section 2.

A. Let $f_6(n)$ denote the expected number of tests under procedure R_I for $t = 2$. From the definition, we easily obtain the recursion formulas for $m \geq 2$:

$$(3.1)\quad \begin{aligned} f_6(2m) &= 2f_6(m) + 2, \\ f_6(2m+1) &= f_6(m) + f_6(m+1) + 2, \end{aligned}$$

with boundary conditions $f_6(2) = 1$ and $f_6(3) = 2\frac{2}{3}$.

From the first equation of (3.1), we obtain by iteration for $n = 2m = 2^r$ and $r \geq 1$

$$(3.2)\quad f_6(2^r) = 3 \times 2^{r-1} - 2.$$

For $n = 2^r + c$ (with $0 \leq c < 2^r$), we set $f_6(n) = 3 \times 2^{r-1} - 2 + g_6(c) + k \times 2^r$ in (3.1). After using one boundary condition to show that $k = 0$, we obtain the simpler homogeneous formulas:

$$(3.3)\quad \begin{aligned} g_6(2c) &= 2g_6(c), \\ g_6(2c+1) &= g_6(c) + g_6(c+1), \end{aligned}$$

with only one boundary condition: $g_6(1) = \frac{5}{3}$. By iteration in (3.3), we obtain $g_6(c) = \frac{5c}{3}$. Hence, for all $n \geq 2$,

$$(3.4)\quad f_6(n) = f_6(2^r + c) = 3 \times 2^{r-1} - 2 + \tfrac{5c}{3}.$$

Under Procedure R_I, it is curious to note that all randomness can be traced back to $n = 3$.

Let $\overline{f_6}(n)$ denote the maximum number of tests required under R_I for $t = 2$. The equations for $\overline{f_6}(n)$ are exactly the same as in (3.1), the only change being that the second boundary condition is now $\overline{f_6}(3) = 3$. Repeating the above argument gives $\overline{g_6}(c) = 2c$, and hence for all $n \geq 2$,

$$(3.5)\quad \overline{f_6}(n) = \overline{f_6}(2^r + c) = 3 \times 2^{r-1} - 2 + 2c.$$

Since $f_6(n) \geq \frac{3n}{2} - 2$ and we later exhibit procedures of asymptotic ($n \to \infty$) order $n + \log(n)$, it follows that R_I is asymptotically inefficient. Moreover, several of the other procedures are uniformly E-better (i.e., equal to or smaller in expectation) than R_I for all $n \geq 2$. Similar remarks hold for the maximum length.

B. Let $f_5(n) = E\{T|R_P\}$ for $t = 2$. Since the j^{th} player ($j \geq 3$) wins his first game (and hence plays an 'extra' game) with a probability of $\frac{2}{j}$, it follows that for all $n \geq 2$,

$$(3.6)\quad f_5(n) = n - 1 + \sum_{j=3}^{n} \frac{2}{j} = n - 4 + 2\sum_{j=1}^{n} \frac{1}{j} \approx n + 2\log_e(n).$$

Clearly, if players $j = 2, 3, \ldots, n$ all win, we obtain the maximum length $\overline{f_5}(n)$; hence, for all $n \geq 2$,

$$(3.7)\quad \overline{f_5}(n) = 2n - 3.$$

Although R_P has a better expectation than R_I, it has a minimax value that is much worse; these results already show up in our table for $n \leq 10$. For all n and asymptotically ($n \to \infty$), we have

$$(3.8)\quad f_5(n) \leq n - 4 + 2\left(\log_e(n) + \gamma + \tfrac{1}{2n}\right),$$

where $\gamma = 0.577\ldots$ is Euler's constant; this can be used to show that $f_5(n)$ is smaller than $\frac{3n}{2} - 2$ and hence smaller than $f_6(n)$ for all $n \geq 2$.

C. Let $f_4(n) = E\{T|R_M\}$ for $t = 2$. Let $n = 2^{r_1}+2^{r_2}+\ldots+2^{r_s}$ in binary notation; this partitions the n players at random into s 'connected' subsets of sizes 2^{r_i} ($i = 1, 2, \ldots, s$) with $r_1 > r_2 > \ldots > r_s \geq 0$. Inside these sets, we need a total of $n - s$ comparisons to find the s best players, and between the s subsets, we need an additional $s-1$ comparisons to find the overall best player. The winner of the j^{th} subset has a probability of $\frac{2^{r_j}}{n}$ of being the overall best. Since we do a knock-out tournament within each subset and because of the order in step 2, this winner carries along with him r_j contenders for second-best from his own subset, $j - 1$ more from the $j - 1$ larger subsets, and $1 - \delta_{js}$ from the smaller subsets; here, $\delta_{js} = 1$ if $j = s$, and $\delta_{js} = 0$ if $j < s$. Thus, if the j^{th} subset produces the best one, then an additional $r_j + (j - 1)+(1 - \delta_{js}) - 1$ comparisons are needed. Hence, for all $n \geq 2$,

$$\begin{aligned} f_4(n) &= (n-s) + (s-1) + \frac{1}{n}\sum_{j=1}^{s}\left(r_j + j - 1 - \delta_{js}\right)2^{r_j} \\ &= n - 2 + \frac{1}{n}\sum_{j=1}^{s}\left(r_j + j - \delta_{js}\right)2^{r_j}. \end{aligned} \tag{3.9}$$

This is not easily amenable to an asymptotic analysis; we, therefore, derive a lower bound for $f_4(n)$ and use the maximum value as an upper bound. A lower bound is obtained by taking only the first term of the summation in (3.9). We note that $r_1 = \lfloor \log(n) \rfloor$ and that $r_1 - \delta_{1s} = \lfloor \log(n-1) \rfloor$. Hence, for all $n \geq 2$,

$$f_4(n) \geq n - 2 + \frac{2^{\lfloor \log(n) \rfloor}}{n}(1 + \lfloor \log(n-1) \rfloor). \tag{3.10}$$

This already shows that for any sequence n_i of n-values,

$$f_4(n_i) \geq n_i - 2 + \tfrac{1}{2}\log(n_i - 1), \tag{3.11}$$

which puts a lower bound on the possible asymptotic form of $f_4(n_i)$ as $n_i \longrightarrow \infty$.

The maximum $\overline{f_4}(n)$ required under R_M occurs when the winner of the first subset (of size 2^{r_1}) is the overall winner, and hence,

$$f_4(n) \leq \overline{f_4}(n) = (n-s) + (s-1) + r_1 - \delta_{1s} = n - 1 + \lfloor \log(n-1) \rfloor. \tag{3.12}$$

Since this same value was shown by Schreier [19] & Słupecki [20] to be a lower bound for the minimax value of any procedure, it follows that R_M is an M-optimal procedure.

The procedure R_M is also important because it attempts to solve the $t = 2$ problem by separating the two problems of finding the best and (conditional on the extra information picked up) then finding the second best. Although this idea was also used by Picard in Section 7.3.1 of [17], it should be noted that our procedure R_M is not the same as his procedure; call the latter procedure R_{P_1}. In fact, it is fairly easy to show (details are omitted) that for any $n \geq 2$, the procedure R_{P_1} has an expectation of

$$E\left\{T|R_{P_1}\right\} = n - 1 + \tfrac{2}{n}(n-2) + \tfrac{1}{n}\sum_{j=1}^{n-2}(j-1) = \tfrac{3}{2}(n-1) - \tfrac{1}{n} \approx \tfrac{3n}{2}, \tag{3.12a}$$

which is to be compared with the upper bound $n + \log(n)$ obtained for R_M in (3.12) above. We can say that R_{P_1} is inadmissible for both the E-goal and the M-goal since R_M is at least as good for

all n and, in fact, strictly better for $n \geq 4$. In particular, for the example with $n = 5$ considered by Picard, R_{P_1} gives 5.8 and 7 for the expectation and maximum, respectively, compared to 5.6 and 6 for R_M.

D. Let $f_3(n)$ denote $E\{T|R_{I^*}\}$ for $t = 2$. For $n = 2^r + c$ (with $0 \leq c < 2^r$), let $g_r(n)$ denote the probability that there are r contenders for second best after step 2 of the procedure R_{I^*}. To show that $g_r(n) + g_{r+1}(n) = 1$, assume for any $n' < 2^r$ (say, n' associated with $r' < r$) that the number of contenders for second best is either r' or $r' + 1$ with probability one. Then, for even $n = 2m$, as a result of step 3,

$$(3.13)\quad \begin{aligned} g_r(2m) &= g_{r-1}^2(m) + g_{r-1}(m)g_r(m) = g_{r-1}(m),\\ g_{r+1}(2m) &= g_r^2(m) + g_{r-1}(m)g_r(m) = g_r(m), \end{aligned}$$

and the sum of these two equations is again one. Also, for odd $n = 2m + 1$,

$$(3.14)\quad \begin{aligned} g_r(2m+1) &= g_{r-1}(m)g_{r-1}(m+1) + g_{r-1}(m)g_r(m+1)\left(\tfrac{m}{2m+1}\right)\\ &\qquad + g_r(m)g_{r-1}(m+1)\left(\tfrac{m+1}{2m+1}\right),\\ g_{r+1}(2m+1) &= g_r(m)g_r(m+1) + g_{r-1}(m)g_r(m+1)\left(\tfrac{m+1}{2m+1}\right)\\ &\qquad + g_r(m)g_{r-1}(m+1)\left(\tfrac{m}{2m+1}\right), \end{aligned}$$

and the sum is $(g_{r-1}(m) + g_r(m))(g_{r-1}(m+1) + g_r(m+1)) = 1$. Since $g_1(2) = 1$, and $g_2(2) = 0$, this result must hold for all $n \geq 2$.

Since r is determined by m, we now write $g(m)$ without the subscript r and obtain from (3.13) and (3.14) (and the result just proved)

$$(3.15)\quad \begin{aligned} g(2m) &= g(m),\\ g(2m+1) &= \begin{cases} \frac{mg(m)+(m+1)g(m+1)}{2m+1} & \text{if } m+1 \text{ is not a power of 2},\\ \frac{mg(m)}{2m+1} & \text{if } m+1 \text{ is a power of 2}, \end{cases} \end{aligned}$$

where $g(2) = 1$, and $g(3) = \frac{1}{3}$. It is easily checked that for $2^r \leq n < 2^{r+1}$ and $r \geq 1$, the solution is

$$(3.16)\quad g(n) = \frac{2^{r+1}-n}{n}.$$

Since it takes exactly $n - 1$ comparisons to find the best and an additional $r - 1$ or r comparisons with probabilities $g(n)$ and $1 - g(n)$, respectively, for the second best, we have from (3.16) for $2^r \leq n < 2^{r+1}$ and $r \geq 1$

$$(3.17)\quad \begin{aligned} f_3(n) &= n - 1 + (r-1)g(n) + r(1 - g(n)) = n + r - \frac{2^{r+1}}{n}\\ &= n + \lfloor \log(n) \rfloor - \frac{2^{1+\lfloor \log(n) \rfloor}}{n}. \end{aligned}$$

The smallest value we add to $n - 1$ in the above is $r - 1 = \lfloor \log(n) \rfloor - 1$, and the largest is $r - \delta_{1s} = \lfloor \log(n-1) \rfloor$; hence,

$$(3.18)\quad n - 2 + \lfloor \log(n) \rfloor \leq f_3(n) \leq n - 1 + \lfloor \log(n-1) \rfloor.$$

Thus, $f_3(n)$ is of asymptotic ($n \to \infty$) form $n + \log(n)$. By the same argument as in (3.18), the maximum length is

$$(3.19)\quad \overline{f_3}(n) = n - 1 + \lfloor \log(n-1) \rfloor.$$

Since the minimax value has to be at least this by Schreier's result, it follows that procedures R_{I^*} and R_M are both M-optimal.

E. Let $f_2(n) = E\{T|R_F\}$ for $t = 2$; for this procedure, we start with the minimax problem and $\overline{f_2}(n)$. It follows directly from the details of step 3 (see Section 2) that according to if n is even ($2m$) or odd ($2m+1$), respectively, we have

$$(3.20)\quad \begin{aligned} \overline{f_2}(2m) &= \overline{f_2}(m) + m + 1, \\ \overline{f_2}(2m+1) &= \overline{f_2}(m) + m + 3, \end{aligned}$$

where $\overline{f_2}(2) = 1$, and $\overline{f_2}(3) = 3$. Letting $g(n) = \overline{f_2}(n+1) - \overline{f_2}(n)$ and setting $\overline{f_2}(1) = -1$ gives

$$(3.21)\quad \begin{aligned} g(2m) &= 2, \\ g(2m+1) &= g(m) - 1, \end{aligned}$$

where $g(1) = 2$ is the only boundary condition. Clearly,

$$(3.22)\quad g(4m+1) = g(2m) - 1 = 1 = g(3).$$

If $m = 2c + 1$ is an odd integer, then by (3.21) and (3.22),

$$(3.23)\quad g(4m-1) = g(8c+3) = g(4c+1) - 1 = 0 = g(7).$$

In general, if $m = 2^{p-2}d$ where d is odd and $p \geq 2$, then by iteration and (3.23),

$$(3.24)\quad g(4m-1) = g(2^p d - 1) = g\big(2^{p-1}d - 1\big) - 1 = g(4d-1) - (p-2) = 2 - p.$$

A single expression for $\overline{f_2}(n)$ for both odd and even n can now be obtained by summing the values $g(j)$ ($j = 1, 2, \ldots, n-1$) and $\overline{f_2}(1)$. In a straightforward manner, we obtain

$$(3.25)\quad \begin{aligned} \overline{f_2}(n) &= n - \tfrac{1+(-1)^n}{2} + \sum_{j=1}^{\lfloor \log(n/2) \rfloor} (2-j) \left\lfloor \tfrac{n+2^j}{2^{j+1}} \right\rfloor \\ &= \left\lfloor \tfrac{5n-(-1)^n}{4} \right\rfloor - \sum_{j=1}^{\lfloor \log(n/8) \rfloor} j \left\lfloor \tfrac{n+2^{j+2}}{2^{j+3}} \right\rfloor. \end{aligned}$$

It is not clear how to show that this is of asymptotic form $n + \log(n)$ because of the appearance of $(\log(n))^2$ in the asymptotic analysis. However, for $n = 2^r$, it is easy to show (we omit the details) that $\overline{f_2}(2^r) = 2^r + r - 2$. It follows from Schreier's result that for $n = 2^r$, we need at least

$$(3.26)\quad n - 1 + \lfloor \log(n-1) \rfloor = 2^r - 1 + \lfloor \log(2^r - 1) \rfloor = 2^r + r - 2$$

comparisons, and hence, it follows that procedure R_F (for $t = 2$) is M-optimal for n equal to a power of 2.

We note from the table that procedure R_F has a slight inefficiency for $n = 7 = 2^3 - 1$. This gets magnified for $n = 15, 31$, and 63, and it is quite surprising to find that $\overline{f_2}(n)$ is not monotonic; in fact, $\overline{f_2}(15) = 19 > \overline{f_2}(16) = 18$. This means that in a tournament with $n = 15$ players, it would be better (in the minimax sense) to introduce a fictitious 16th player (say, a beneficent deity) who always loses and hence never is selected to be best or second best. For $n = 62$, we could use two such deities since $\overline{f_2}(62) = 69$, $\overline{f_2}(63) = 71$, and $\overline{f_2}(64) = 68$. This lack of monotonicity did not occur with our previous procedures and is conjectured not to occur for any of the entropy procedures. It also serves to prove that R_F is not M-optimal for $t = 2$, and as our table shows, it is also not E-optimal. However, the analogous procedure for the complete ranking problem ($t = n-1$) is quite efficient and was shown [8] to be M-optimal for $n \leq 11$ and for $n = 20, 21$; S. Johnson (personal communication) states that it was also shown by M. Wells by machine methods to be M-optimal for $n = 12$.

For both $t = 2$ and $t = n-1$, the procedure R_F is also of interest for its relatively low expectation. To find an exact expression for the expectation $f_2(n)$, we return to step 3 for odd n $(2k+1)$ and compute the probabilities associated with the tree for R_F in Section 2. The total number of equally likely cases after step 2 is

(3.27) $C = \binom{4}{2}\binom{6}{2}\dots\binom{2k-4}{2}(2k-3)(n-2)n = \frac{(2k+1)!}{k(k+1)2^k}$.

For the first comparison (after step 2), these are split into

(3.28) $C_1 = \binom{4}{2}\dots\binom{2k-4}{2}(2k-3)(n-3)(n-1)$ and $C - C_1$

cases for the left and right fork, respectively, thus yielding the probabilities

(3.29) $p_1 = \frac{(n-3)(n-1)}{n(n-2)}$ and $p_2 = \frac{2n-3}{n(n-2)}$.

Similarly, the two probabilities for the one remaining fork in our tree for R_F (in Section 2) are easily computed to be

(3.30) $p_{21} = \frac{n-1}{2n-3}$ and $p_{22} = \frac{n-2}{2n-3}$.

Hence, the expectation associated with step 3 for n odd is

(3.31) $2(p_1 + p_2 p_{22}) + 3p_2 p_{21} = 2 + \frac{n-1}{n(n-2)} < 3$,

instead of the three used in the 2nd equation of (3.20). Thus, we have to subtract $1 - \frac{n_i-1}{n_i(n_i-2)}$ for each odd integer $n_i \geq 3$ that appears in the sequence n, $\lfloor \frac{n}{2} \rfloor$, $\lfloor \frac{n}{4} \rfloor$, ...; suppose there are t such integers $n_1, n_2, \dots, n_t$. Then, our result is

(3.32) $f_2(n) = \overline{f_2}(n) - t + \sum_{i=1}^{t} \frac{n_i-1}{n_i(n_i-2)}$,

where $\overline{f_2}(n)$ is given by (3.25).

Although it is not proved that $f_2(n)$ is strictly increasing in n, this does appear to be true by the table in Section 2 and by specific calculations for $n = 15, 16, 62, 63$, and 64. In particular, we note that for $n = 2^r$, we obtain from (3.32)

(3.33) $f_2(2^r) = \overline{f_2}(2^r) = 2^r + r - 2$.

It has not been proved that $2^r + r - 2$ is a lower bound for $E\{T|n = 2^r\}$ for all procedures, but this is conjectured to be true. It is not too difficult to show that this lower bound holds among all procedures in certain classes (e.g., the class with the property that the best one is selected in the first $n-1$ comparisons), but the general result is still outstanding.

It can be shown that the procedure R_F is the best one given that the first two steps of R_F are to be used, namely ordinary pairing and (semi-) induction on the winners; such results are considered by Hadian [10].

F. Let $f_1(n) = E\{T|R_E\}$ for $t = 2$. For the entropy procedures, we have no exact formulas for all n and hence less complete results. The major evidence of the efficiency of these procedures lies in the numerical results and comparisons. We describe in some detail the procedure R_E for $n = 6$. The table in Section 2 shows that for $n \leq 10$, our best results are consistently obtained by one of the three entropy procedures. In particular, R_{E_1} appears to be the best of all.

Without exact formulas, we cannot prove that the expectation under R_E has the same asymptotic form $n + \log(n)$ as under procedure R_{I^*}, but this is conjectured to be true. In the next section, we derive lower bounds for the expectation under R_{E_1} and R_{E_2}. In the table in Section 2, there are given values of $n - 2 + \frac{1}{2}\lfloor 2\log(n)\rfloor$, which is conjectured to be a lower bound for all procedures for $t = 2$.

For $n = 6$, we now illustrate in detail one step in the calculations for R_E. It was previously found that the procedure tests 1 vs. 2 and (for all $n \geq 4$) then 3 vs. 4 and then (assuming even numbers are the winners) 2 vs. 4. After this, a complete pairing (defined below) procedure tests 5 vs. 6, and as shown in the next section, this reduces the entropy by $\frac{2(2n-3)}{n(n-1)}$, which equals $\frac{6}{10}$ for $n = 6$. We wish to show that this test is not used by R_E since the test 5 vs. 2 gives a larger reduction in entropy. The expected uncertainty $E\{U\}$ after 2 vs. 4 (assuming 4 & 2 are the winners) is easily shown by direct calculation or by (4.7) and (4.8) below, to be

$$E\{U\} = \log(30) - \tfrac{32}{15} = 2.773\ldots \tag{3.34}$$

The probability that 5 loses to 2 (resp., wins over 2) at this stage after 4 beats 2 is easily seen to be $\frac{8}{15}$ (resp., $\frac{7}{15}$).

If 5 loses to 2, then we are left with the following sets of possible (true) states of nature:

- 1 subset (called D_2^4) with 24 cases,
- 3 subsets (called D_3^4, D_4^6, and D_6^4) with 8 cases each.

The total number of cases for the left fork is 48.

If 5 wins over 2, then we are left with the cases:

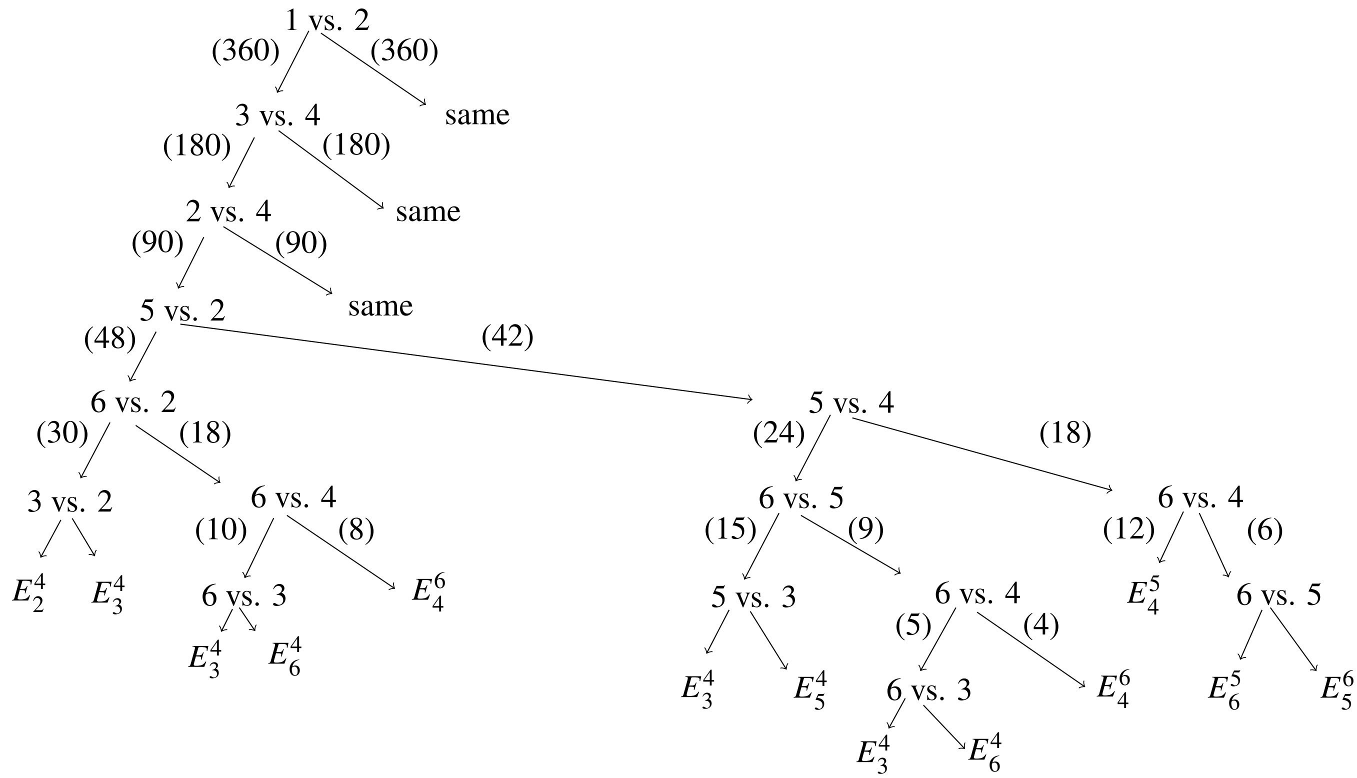
- 2 subsets (called D_4^5 and D_5^4) with 12 cases each,
- 3 subsets (called D_3^4, D_4^6, and D_6^4) with 4 cases each,
- 2 subsets (called D_5^6 and D_6^5) with 3 cases each.

The total number of cases for the right fork is 42. Here, D_i^j indicates the possible decision that j is best and i is second best. Hence, the expected uncertainty after 5 vs. 2 is

$$\begin{aligned} E\{U\} &= \tfrac{8}{15}\Big(\tfrac{1}{2}\log(2) + \tfrac{1}{2}\log(6)\Big) + \tfrac{7}{15}\Big(\tfrac{4}{7}\log\Big(\tfrac{7}{2}\Big) + \tfrac{2}{7}\log\Big(\tfrac{21}{2}\Big) + \tfrac{1}{7}\log(14)\Big) \\ &= \tfrac{1}{5} + \tfrac{2}{5}\log(3) + \tfrac{7}{15}\log(7) = 2.144\ldots \end{aligned} \tag{3.35}$$

Hence, the reduction in entropy is the difference $\left(\log(30) - \frac{32}{15}\right) - \left(\frac{1}{5} + \frac{2}{5}\log(3) + \frac{7}{15}\log(7)\right) = 0.629\ldots$, which is greater than the reduction 0.6 obtained by the test 5 vs. 6. This result only held for $n = 6$, and in fact, 5 vs. 6 gives a bigger reduction in entropy for all $n \geq 7$.

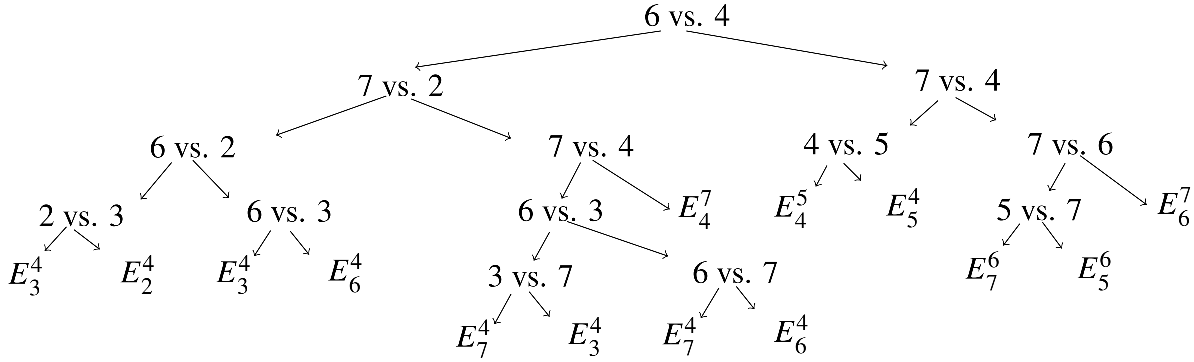
The final tree obtained for $n = 6$ under R_E is:



Here, the word 'same' indicates a repetition of the corresponding left fork. The numbers in parentheses show the partition of the original $6! = 720$ cases or states of nature and are useful in computing the expectation. The symbol E_i^j indicates an endpoint where the decision D_i^j, that j is best and i is second best, is made.

No other procedure was found that had a smaller expectation for $n = 6$, but three of our procedures have a maximum length of 7.

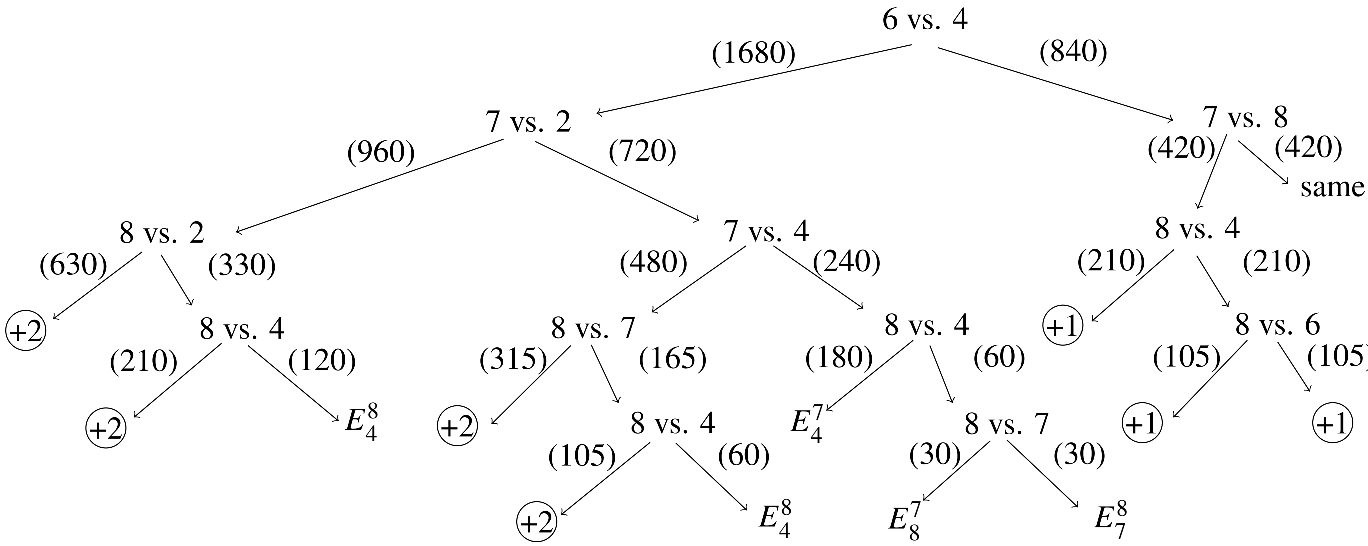
For $n \leq 4$, the entropy procedures are the optimal procedures in common with 3 of the other procedures. For $n = 5$, they coincide with the procedure R_F, giving an expectation of $5\frac{4}{15}$ and a maximum length of 6. For $n = 7$, we use complete pairing, i.e., 1 vs. 2, 3 vs. 4, 2 vs. 4, and 5 vs. 6. By our convention, the number with the higher power of 2 is the winner. The rest of R_E (as well as R_{E_1} and R_{E_2}) is given by:



Here, the expectation is $4 + 3\frac{17}{21} = 7\frac{17}{21}$, and the maximum length is $4 + 5 = 9$.

It is interesting to note that R_E tests 6 vs. 4 (after 1 vs. 2, 3 vs. 4, 2 vs. 4, and 5 vs. 6) for all $n \geq 7$, whereas R_{E_1} and R_{E_2} both test 7 vs. 8 (and then 6 vs. 8 and then 4 vs. 8) for all $n \geq 8$. Hence, for $n \geq 8$, the procedure R_E differs from both R_{E_1} and R_{E_2} and is more difficult to obtain.

For $n = 8$, the continuation for the procedure R_E (after 1 vs. 2, 3 vs. 4, 2 vs. 4, and 5 vs. 6) was found to be:



Here, the symbol $(+j)$ denotes that exactly j more obvious comparisons are needed to complete the procedure. In this instance, the expectation is readily computed to be $9\frac{1}{168}$, and the maximum length is 11.

For $n = 8$, the procedure R_{E_1} (which is the same as R_{E_2} for $n = 2^r$, any integer r) only requires 9 comparisons on average and has a maximum length of 10. It is conjectured that procedure R_{E_1} will continue to be as good or better than R_E for all larger values of n.

4. Cycle-Pairing, Complete Pairing and Ordinary Pairing

Ordinary pairing means, of course, that k comparisons are made when $n = 2k$ or $2k + 1$. A knock-out tournament for getting the best player when $n = 2^r$ consists of an ordinary pairing of all those players that won in the previous round. Hence, the number of rounds is r, and the total number of comparisons is $n - 1$. To define complete and cycle pairing, we make use of a lemma.

Lemma: If the highest power of 2 that factors into $n!$ is p, i.e., $n! = 2^p(2c + 1)$ with $c \geq 0$ an integer, and the integer s is defined by writing n in binary notation as

(4.1) $n = 2^{r_1} + 2^{r_2} + \ldots + 2^{r_s}$,

where $r_1 > r_2 > \ldots > r_s \geq 0$, then

(4.2) $p = n - s = \sum_{i=1}^{s} (2^{r_i} - 1) = \sum_{j=1}^{r_1} \lfloor \frac{n}{2^j} \rfloor$.

Proof: Using induction on n, the inference from n to $n+1$ for even n is obvious since p is not changed and n (resp., s) increases by one. If n is odd, then $r_s = 0$. Suppose $n + 1$ replaces $1 + 2 + \ldots + 2^{j-1}$ with 2^j. Then p is increased by j, s is decreased by $j - 1$, and n is, of course, increased by one.

Since the result also holds for $n = 1$, the result is proved. The proof of the last equality in (4.2) is omitted. From the first summation in (4.2), we see that p is exactly the number of comparisons needed to find the best player in each of the s subsets of sizes 2^{r_i} ($i = 1, 2, \ldots, s$).

For complete pairing, we form the s subsets defined by (4.1) and perform the p comparisons needed to find the best player in each subset; this type of pairing is used in R_M and R_{E_2}. For cycle pairing, we only do the $2^{r_1} - 1$ pairings needed to find the best player in the largest subset of size 2^{r_1}, as defined in (4.1). Of course, for $n = 2^r$, these two concepts coincide.

Since we are conjecturing that among the E-optimal procedures, there is a cycle-pairing procedure, it is of interest to let R_C denote any cycle-pairing procedure and see what properties it has; this is the aim of the present section.

For $n = 2^r$, the cycle-pairing (as well as the complete-pairing) procedure gives us, after $n - 1$ comparisons, the best player and exactly r contenders for second best. Since we need exactly $r - 1$ further comparisons to find the second best, it follows that for any procedure R_C with $n = 2^r$,

(4.3) $E\{T|R_C\} = 2^r + r - 2 = \max\{T|R_C\}$.

We now obtain a lower bound for each of the three types of pairing. If there is a cycle pairing procedure among the E-optimal procedures, then the lower bound for any R_C should also be a lower bound for all procedures.

We define a comparison C_j [or C_j (a vs. b)] to be of level j if the two players, a and b, each have $2^{j-1} - 1$ inferiors, the two sets are disjoint, and each of the two players has no proven superiors ($j = 1, 2, \ldots, \lfloor \log(n) \rfloor$). We want to prove a result about the reduction in entropy for any comparison of level j, regardless of where it occurs in our procedure. First, take $j = 1$; consider C_1 (a vs. b) and assume that we may or may not have some incomplete knowledge from comparisons among the remaining $n - 2$ players.

Lemma: The reduction in entropy r_1 (a vs. b) due to the 1st level comparison C_1 (a vs. b) is given by

(4.4) r_1 (a vs. b) $= \frac{2(2n-3)}{n(n-1)}$,

regardless of the knowledge previously obtained about ordering that affects only the remaining $n - 2$ players.

Proof: For convenience, take $a = 2k + 1$ and $b = 2k + 2 \leq n$ and assume the previous knowledge concerns only players $1, 2, \ldots, 2k$. Consider any definite order, say $1 \prec 2 \prec \ldots \prec 2k$, for these $2k$ players (where $\prec$ means 'is inferior to'); the same argument holds for any such fixed order. The remaining subsets of possible states of nature corresponding to the possible decisions D_i^j (i, $j \geq 2k$), D_i^{2k}, D_{2k}^i ($i > 2k$), and D_{2k-1}^{2k} and the number of cases (or the relative probability) for each before the comparison C_1 is made are as follows:

$$\begin{aligned} &(n-2k)(n-2k-1) \text{ subsets with } \tfrac{(n-2)!}{(2k)!} \text{ cases in each,} \\ (4.5) \qquad &2(n-2k) \text{ subsets with } \tfrac{(n-2)!}{(2k-1)!} \text{ cases in each,} \\ &1 \text{ subset with } \tfrac{(n-2)!}{(2k-2)!} \text{ cases in it.} \end{aligned}$$

The probability of each subset is simply the number of cases in it divided by the total number of possible cases; by (4.5), this total is $\frac{n!}{(2k)!}$. Hence, the uncertainty $E_1\{U\}$ before making the comparison C_1 is

$$
\begin{aligned}
E_1\{U\} = {} & \tfrac{(n-2k)(n-2k-1)}{n(n-1)} \log(n(n-1)) + \tfrac{(2n-4k)2k}{n(n-1)} \log\left(\tfrac{n(n-1)}{2k}\right) \\
& + \tfrac{2k(2k-1)}{n(n-1)} \log\left(\tfrac{n(n-1)}{2k(2k-1)}\right) = \log(n(n-1)) - \tfrac{2k(2n-2k-1)}{n(n-1)} \log(2k) \\
& - \tfrac{2k(2k-1)}{n(n-1)} \log(2k-1).
\end{aligned} \tag{4.6}
$$

After making the comparison C_1 and assuming by our convention that $2k+1$ loses to $2k+2$, the subsets in the first two rows of (4.5) which put $2k+1$ in the first or second place have to be treated separately, and we then have the five types:

$$
\begin{aligned}
& 2n-4k-3 \text{ subsets with } \tfrac{(n-2)!}{(2k)!} \text{ cases in each,} \\
& (n-2k-2)(n-2k-3) \text{ subsets with } \tfrac{(n-2)!}{2(2k)!} \text{ cases in each,} \\
& 2 \text{ subsets with } \tfrac{(n-2)!}{(2k-1)!} \text{ cases in each,} \\
& 2n-4k-4 \text{ subsets with } \tfrac{(n-2)!}{2(2k-1)!} \text{ cases in each,} \\
& 1 \text{ subset with } \tfrac{(n-2)!}{2(2k-2)!} \text{ cases in it.}
\end{aligned} \tag{4.7}
$$

From (4.7), we find that the total number of cases is $\frac{n!}{2(2k)!}$. Since the complementary result, $2k+1$ wins over $2k+2$, gives rise to a symmetrical set of results, it follows that the expected uncertainty $E_2\{U\}$ after the comparison C_1 can now be obtained from (4.7). By straightforward algebra, as in (3.35), and subtraction from (4.6), we obtain the desired result:

$$E_2\{U\} - E_1\{U\} = \frac{2(2n-3)}{n(n-1)}. \tag{4.8}$$

If we average this result over various possible fixed values of $1, 2, \ldots, 2k$, then we obtain the same result (4.7) for any partial knowledge about the players $1, 2, \ldots, 2k$ (or any subset thereof), and this proves our result.

A similar calculation for any comparison C_j of level j ($j = 1, 2, \ldots, \lfloor \log(n) \rfloor$) gives the more general result:

$$r_j\ (a \text{ vs. } b) = \frac{2^j\left(2n-1-2^j\right)}{n(n-1)}. \tag{4.9}$$

Since the proof is quite similar to the lemma above, we omit the proof of (4.9) for $j > 1$.

For $n = 2^r + c$ ($0 \le c < 2^r$), any cycle-pairing procedure R_C has at least 2^{r-1} pairings of level 1, at least 2^{r-2} pairings of level 2, and so on, to at least one pairing of level r. We can assume that it has exactly these numbers of pairings among the first $2^r - 1$ comparisons. Then, the reduction in entropy due to these comparisons is, using (4.9),

$$Q = \sum_{j=1}^{r} \frac{2^j\left(2n-1-2^j\right)2^{r-j}}{n(n-1)} = 2^r\left(\frac{(2n-1)r-2^{r+1}+2}{n(n-1)}\right). \tag{4.10}$$

Let $T_1 = 2^r - 1$ denote the number of these comparisons, and T_2 denote the remaining so that $T = T_1 + T_2$. Since the total uncertainty at the outset is $\log(n(n-1))$, and 1 is an upper bound for the reduction in entropy for all steps (in particular, for those after the first T_1), it follows that

$$Q + (1 \times E\{T_2\}) \ge \log(n(n-1)). \tag{4.11}$$

Hence, with the help of (4.11), we obtain the desired lower bound for any cycle-pairing procedure R_C:

(4.12) $E\{T|R_C\} = 2^r - 1 + E\{T_2\} \geq 2^r - 1 + \log(n(n-1)) - 2^r\left(\frac{(2n-1)r-2^{r+1}+2}{n(n-1)}\right).$

Of course, for $n = 2^r$, we obtain an improvement by using (4.3).

The corresponding result for complete pairing is obtained by using (4.1) and noting that the first $p = n - s$ comparisons consist of $\lfloor \frac{n}{2^j} \rfloor$ comparisons C_j of level j ($j = 1, 2, \ldots, r_1$). Hence, we replace Q in (4.11) with

(4.13) $Q_1 = \sum_{j=1}^{r_1} \frac{2^j\left(2n-1-2^j\right)}{n(n-1)} \lfloor \frac{n}{2^j} \rfloor \leq \frac{(2n-1)r_1 - 2^{r_1+1}+2}{n-1}.$

By a similar argument to that above, we have for any procedure that uses complete pairing (such as R_M)

(4.14) $E\{T\} \geq n - s + \log(n(n-1)) - \sum_{j=1}^{r_1} \frac{2^j\left(2n-1-2^j\right)}{n(n-1)} \lfloor \frac{n}{2^j} \rfloor,$

where the sum can be bounded as in (4.13) for asymptotics; here again, we get an improvement for $n = 2^r$ by using $2^r + r - 2$ from (4.3).

For ordinary pairing, we use $\lfloor \frac{n}{2} \rfloor$ pairings of level 1 only and find in a similar manner that for any procedure that uses ordinary pairing (such as R_F),

(4.15) $E\{T\} \geq \log(n(n-1)) + \frac{(n-2)(n-3)}{n(n-1)} \lfloor \frac{n}{2} \rfloor.$

Since ordinary pairing and cycle pairing are both part of the complete pairing, it follows that the lower bound in (4.14) is not less than those in (4.12) and (4.15). However, since we conjecture that there is a cycle-pairing procedure among those that are E-optimal, the lower bound in (4.12) is of more interest; it is given in the table in Section 2 without the improvement for $n = 2^r$.

Although the lower bound in (4.12) is asymptotically ($r \to \infty$) equal to n for $n = 2^r$, it should be pointed out that for $n = 3 \times 2^{r-1}$, the asymptotic ($r \to \infty$) value is only $\frac{2}{3}(n + \log(n)) + O(1)$.

5. Procedures for the Ordering Problem with $t = n - 1$

Several procedures are introduced, all of which are new, except for procedure R_S due to Steinhaus [23] and R_F due to Ford and Johnson [8]. Our main interest is in the concept of the maximum expected reduction in entropy in g steps for small positive integers g. It is shown in Section 6 that for $g = 1$, this maximum is achieved by finding the comparison that partitions all the remaining possible states of nature (or cases) into two sets that are (as close as possible to being) equal in size. For the g-step (expected) entropy procedure, we wish to make the 2^g subsets (as far as possible) equal in size in the sense of maximizing $-(p_1 \log(p_1) + p_2 \log(p_2) + \ldots + p_{2^g} \log(p_{2^g}))$ where $p_i = \frac{C_i}{T}$, where C_i ($i = 1, 2, \ldots, 2^g$) is the number of cases in the i^{th} subset, and $T = C_1 + C_2 + \ldots + C_{2^g}$ is the total number of cases. The concept of complete pairing (explained in Sections 2 and 4) also enters all of the new procedures. The word 'expected' in referring to entropy procedures is dropped after Section 5.

The procedure R_N uses the idea of inserting units into a 'main chain,' and it changes the unit to be inserted when there is evidence that 'noise' is entering the procedure. The concept of noise, the criterion for noticing its presence, and its relation to the expectation are discussed in Section 6.

Procedure R_{E^*}: This is essentially a 1-step entropy procedure for $t = n - 1$, i.e., it is based on finding the binary comparison that maximizes the expected reduction in entropy after one comparison. At some isolated points, we allow the use of 2-step or 3-step entropy without a formalized rule explaining when the higher-step entropies will be used. Complete pairing is used for the first p comparisons.

Procedure R_E: This is a pure 1-step entropy procedure, which also uses complete pairing for the first p comparisons. Higher-step entropies are used to decide between two comparisons only when the 1-step entropy reductions are equal.

Procedure R_N: This procedure also uses complete pairing for the first p comparisons; this establishes a 'main chain' (denoted by the powers of 2 under our convention). After that, units are inserted in the main chain, i.e., we only compare a unit off the main chain with a unit on the main chain. We continue to try to insert the unit chosen until either it is inserted or there is evidence that noise (denoted by N) is entering the procedure (a criterion for this is given). The decision as to which unit should be inserted and what comparison to make is sequential and based on 1-step entropy considerations, i.e., given the present state of knowledge, we select the comparison that maximizes the expected reduction in entropy due to the next comparison only.

It should be clear from the above procedures that further improvement through the use of higher-step entropies is thought to be possible, but this requires extra computation and has not been investigated.

Procedure R_G: This procedure is based on first ordering separately the s subsets formed by complete pairing and then using the 1-step entropy criterion for merging these ordered subsets, each of size equal to a power of 2. To get something different than R_E for $n = 2^r$, we assume that each of the two halves of size 2^{r-1} must be separately ordered and then merged.

The table below shows the numerical results for these procedures and compares them with those for R_S and R_F. Important omissions from this table are the optimal procedures of C. Picard [17] for $n = 6$ and a procedure of Cesari [4] for $n = 7$, which has only three units of noise; no rule for general n is given in their work. The lower bound LB in the table is defined as

(5.1) $\mathrm{LB} = r + \frac{2c}{n!}$,

where r and c are defined by writing $n! = 2^r + c$ $(0 \le c < 2^r)$; this result comes from the work of Huffman [12] and was applied to this problem independently by Kislitsyn [14] and through questionnaire theory by Picard [17]. The corresponding minimax lower bound $\mathrm{MLB} = 1 + \lfloor \log(n!) \rfloor$ for $n \ge 3$ in the minimax problem was used by Ford and Johnson [8] and is also discussed by Steinhaus [25].

Comparison of Six Procedures for the Complete Ordering Problem

Lower Bound and Procedures	Expected Values								
	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
LB	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{26}{45}$	$12\frac{118}{315}$	$15\frac{118}{315}$	$18\frac{1574}{2835}$	$21\frac{11966}{14175}$
R_{E^*}	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{26}{45}$	$12\frac{121}{315}$	$15\frac{121}{315}$	$18\frac{1592}{2835}$	N.C.
R_N	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{26}{45}$	$12\frac{122}{315}$	$15\frac{122}{315}$	$18\frac{1608}{2835}$	N.C.
R_E	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{26}{45}$	$12\frac{123}{315}$	$15\frac{123}{315}$	$18\frac{1624}{2835}$	N.C.
R_F	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{27}{45}$	$12\frac{144}{315}$	$15\frac{144}{315}$	$18\frac{1656}{2835}$	$21\frac{12060}{14175}$
R_G	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$7\frac{1}{15}$	$9\frac{30}{45}$	$12\frac{150}{315}$	$15\frac{177}{315}$	$18\frac{1737}{2835}$	$21\frac{13725}{14175}$
R_S	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$7\frac{1}{15}$	$9\frac{33}{45}$	$12\frac{186}{315}$	$15\frac{186}{315}$	$18\frac{2304}{2835}$	$22\frac{3015}{14175}$

Noise Units (NU) (Noise $N = \frac{\text{NU}}{\text{Column Denominator}}$)

	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
R_{E^*}	0	0	0	0	0	3	3	18	N.C.
R_N	0	0	0	0	0	4	4	34	N.C.
R_E	0	0	0	0	0	5	5	50	N.C.
R_F	0	0	0	0	1	26	26	82	94
R_G	0	0	0	2	4	32	59	163	1759
R_S	0	0	0	2	7	68	68	730	5224

(Min., Max.) of the Number T of Comparisons under R

	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
MLB	1	3	5	7	10	13	16	19	22
R_{E^*}	(1, 1)	(2, 3)	(4, 5)	(6, 7)	(9, 10)	(11, 13)	(14, 16)	(18, 20)	N.C.
R_N	(1, 1)	(2, 3)	(4, 5)	(6, 7)	(9, 10)	(11, 13)	(14, 16)	(17, 20)	N.C.
R_E	(1, 1)	(2, 3)	(4, 5)	(6, 7)	(9, 10)	(11, 13)	(14, 16)	(18, 20)	N.C.
R_F	(1, 1)	(2, 3)	(4, 5)	(6, 7)	(8, 10)	(10, 13)	(13, 16)	(16, 19)	(19, 22)
R_G	(1, 1)	(2, 3)	(4, 5)	(6, 8)	(8, 11)	(11, 14)	(14, 17)	(17, 20)	(20, 23)
R_S	(1, 1)	(2, 3)	(4, 5)	(6, 8)	(8, 11)	(10, 14)	(13, 17)	(16, 21)	(19, 25)

6. Properties and Proof

We define the 'halving procedure' as one that always selects a comparison that makes the resulting two sets of cases (as far as possible) equal in size. Let T denote the total number of possible states of nature at any stage, and let x and $y = T - x$ denote the partition resulting from some comparison.

Lemma 1: The halving procedure and the 1-step entropy procedure are equivalent.

Proof: The reduction in entropy at any stage is given by

$$(6.1)\quad \log(T) - \tfrac{x}{T}\log(x) - \tfrac{y}{T}\log(y) = -\big(p_x \log(p_x) + p_y \log(p_y)\big),$$

where $p_x = \frac{x}{T}$ and $p_y = \frac{y}{T}$. It is well known that the right side of (6.1) is maximized by setting $p_x = p_y$ or $x = y$, and this proves the result.

Of course, if we could always partition the states of nature exactly in half, then we would have an optimal solution. All our difficulties arise from the fact that this halving is not always possible. On the other hand, it is not necessary to partition the set exactly in half to get an optimal breakdown. We now give some results about this point.

Let $H(T)$ denote the expected number of comparisons required when there are T possible states of nature. Let x denote the smaller of the two subset sizes that result from some comparison; suppose we could choose any subset size x we wish. Then, $H(1) = 0$, and for $T \geq 2$,

$$(6.2)\quad H(T) = 1 + \min_{1 \leq x \leq T/2} \big\{\tfrac{x}{T}H(x) + \tfrac{T-x}{T}H(T-x)\big\}.$$

Let $h(y) = yH(y)$. Then (6.2) takes the simpler form

$$(6.3)\quad h(T) = T + \min_{1 \leq x \leq T/2} \{h(x) + h(T-x)\}.$$

Define r and c by writing $T = 2^r + c$, where $0 \leq c < 2^r$. It can be readily proved, as in Lemma 2 of [21], that the minimum in (6.3) is attained at $x = \frac{T}{2}$ and that an exact expression for $H(T)$ for all $T > 0$ is

$$(6.4)\quad H(T) = r + \tfrac{2c}{T} = r + \tfrac{2}{T}(T - 2^r).$$

The following result was found to be quite useful in searching for procedures with less noise, and in particular it is used in the definition of the procedure R_N. It corresponds to Lemma 3 of [21], but it should be noted that because of different boundary conditions, the result is completely different from that in the above-mentioned lemma.

Lemma 2: For any $T \geq 2$, an integer y will yield the minimum in (6.3) if and only if there is no power of 2 strictly between y and $T - y$.

Proof: Let $h(x;T)$ denote the sum in braces in (6.3); because of the symmetry about $x = \frac{T}{2}$, we assume $x \leq T - x$. Consider different possible inequalities between x, $T - x$, and the power of 2 that is closest to their average $\frac{T}{2}$.

Case 1: $2^{r-1} \leq x \leq T - x \leq 2^r$.

Then, letting $r(x)$ denote the r-value for x, $r(x) = r(T-x) = r - 1$, and to check the equality in (6.3), we use (6.4) and compute:

$$(6.5)\quad \begin{aligned} T + h(x;T) &= T + (r-1)x + 2\big(x - 2^{r-1}\big) + (r-1)(T-x) + 2\big(T - x - 2^{r-1}\big) \\ &= rT + 2(T - 2^r) = h(T). \end{aligned}$$

Hence, the minimum in (6.3) is attained for such values of x.

Case 2A: $2^{s-1} \leq x < 2^s$ and $2^r < T - x$ for $1 \leq s \leq r$, and

Case 2B: $2^{s-1} \leq x < 2^s$ and $2^r = T - x$ for $1 \leq s < r$.

Then $r(x) = s - 1$ and $r(T-x) = r$, and a similar computation gives for both Cases 2A and 2B

$$\begin{aligned} T + h(x;T) &= T + x(s-1) + 2\left(x - 2^{s-1}\right) + (T-x)r + 2(T - x - 2^r) \\ &= h(T) + (T - x - 2^r) + (2^s - x)t + 2^s\left(2^t - t - 1\right) > h(T), \end{aligned} \tag{6.6}$$

since $t = r - s \geq 0$, and (hence) $2^t - t - 1 \geq 0$. In Case 2A (resp., Case 2B), strict inequality follows from the fact that $T - x - 2^r > 0$ (resp., $(2^s - x)t > 0$). Hence, the minimum in (6.3) cannot be achieved for such values of x.

Case 3: $2^{s-1} \leq x < 2^s$ and $2^{r-1} < T - x < 2^r$ for $s \leq r - 1$.

Here, $r(x) = s - 1$, $r(T-x) = r - 1$, and $r - s > 0$. As above, we obtain

$$\begin{aligned} T + H(x;T) &= T + x(s-1) + 2\left(x - 2^{s-1}\right) + (T-x)(r-1) + 2\left(T - x - 2^{r-1}\right) \\ &= h(T) + 2^s\left(2^t - t - 1\right) + (2^s - x)t > h(T), \end{aligned} \tag{6.7}$$

since $t = r - s > 0$ and $2^t - t - 1 \geq 0$. Thus, the minimum in (6.3) cannot be achieved for such x values.

Case 4: $x = 2^s$ and $2^{r-1} < T - x < 2^r$ for $s \leq r - 2$.

Here, $r(x) = s$, $r(T-x) = r - 1$, and $t = r - s \geq 2$. As above, we obtain

$$\begin{aligned} T + H(x;T) &= T + s2^s + (T - 2^s)(r-1) + 2\left(T - 2^s - 2^{r-1}\right) \\ &= h(T) + 2^s\left(2^t - t - 1\right) > h(T), \end{aligned} \tag{6.8}$$

since $2^t - t - 1 > 0$ for $t \geq 2$; the minimum in (6.3) is again not achieved. Since these four cases exhaust the possible relations between x, $T - x$, and the power of 2 closest to their average $\frac{T}{2}$, the lemma is proved.

It follows from this lemma that in selecting a comparison at any stage of a procedure, we can determine, by looking at the two resulting subset sizes (and their relation to the power of 2 closest to their average), whether or not this particular comparison is introducing an inefficiency (which we call noise) into the procedure. This is exactly the criteria that was used in the procedure R_N. It should be mentioned that Lemma 2 is related to the theorem of Sandelius [18], which uses a different approach and does not get our later results.

We are also interested in the amount of noise brought into the procedure, especially when there is exactly one power of 2 strictly between the two subset sizes. For Cases 2A, 2B, 3, and 4, this corresponds to $t = 0$, 1, 1, and 2, respectively. For Case 2A, the amount added to $h(T)$ is $T - x - 2^r$ and

(6.9) $T - x - 2^r < 2^s - x$ since $s = r$ and $T < 2^{r+1}$.

For Cases 2B and 3, the amount added to $h(T)$ is $2^s - x$, and

(6.10) $2^s - x \leq T - x - 2^s$ since $s = r - 1$ and $T \geq 2^r$.

For Case 4, the amount added to $h(T)$ is $2^s = 2^{r-2}$, and

(6.11) $2^{r-2} < T - x - 2^{r-1}$ since $x = 2^{r-2}$ and $T > 2^{r-1}$.

Hence, we have proved the following:
Lemma 3: The noise N due to a comparison with exactly one power of 2 strictly between the subset sizes, $T_1 < 2^a < T_2 = T - T_1$, is simply the minimum distance to this power of 2, i.e.,

$$(6.12)\quad N = \min(2^a - T_1, T_2 - 2^a).$$

The contribution of this noise N to the expectation, if we start with T cases, is then $\frac{N}{T}$; if we start with any larger number D of cases ($D > T$), then this contribution is to be multiplied by the probability $\frac{T}{D}$ of entering this part of the tree. Hence, the overall contribution to the expectation for this arbitrary comparison is $\frac{N}{D}$. This latter result, which we just proved, can be regarded as a corollary to Lemma 3, but its usefulness is such that we prefer to write it as a theorem below. Let the noisy nodes of a tree have noises N_1, N_2, …, N_w; we call a noisy node simple if the two subset sizes obtained by that comparison have exactly one power of 2 between them. The common expected value of any noiseless tree (i.e., one with no noisy nodes) that starts with n possible states of nature is $H(n)$. Then, we have the following theorem.
Theorem: For any procedure R that has only noiseless nodes and simple noisy nodes, the expectation is given by

$$(6.13)\quad E\{T|R\} = H(n) + \frac{1}{n}\sum_{i=1}^{w} N_i,$$

where $H(n)$ is given in (6.4), and the N_i are given by (6.12). This result enables one to keep track of the expectation of a procedure (or the expected length of the tree) while the procedure is still being constructed. Clearly, it is quite useful in searching for the existence or non-existence of noiseless trees. It was used for most of our computations in the table above and also in the footnotes of the procedures listed below.

The above analysis is of general interest to our search problem and is not to be associated only with the entropy procedures. For example, the formula in (6.4) also applies to the Steinhaus procedure R_S. Since the Steinhaus procedure makes the individual insertions without noise, it follows that $H(i)$ is the expected number of comparisons necessary to insert an item into a chain of length i. It easily follows, using (6.4), that the expectation under R_S for n units with $2^r \le n < 2^{r+1}$ is given by

$$(6.14)\quad E\{T|R_S\} = \sum_{i=2}^{n} H(i) = r(n+1) + 2(n - 2^r) - \sum_{j=2}^{n} \frac{2^{1+\lfloor \log(j) \rfloor}}{j}.$$

A similar expression was obtained by Trybula (personal communication); the asymptotic properties have been investigated by Kislitsyn [14] and Hadian [10].

The procedure R_F was defined in [8] and developed by means of separate recursion formulas for odd and even values of n, which involve complicated sums; no explicit expression for the minimax integer $U(n)$ under R_F was given. A single explicit expression for $U(n)$ for all $n \ge 1$ is

$$(6.15)\quad U(n) = j\left(n + \tfrac{1}{2}\right) - \tfrac{4}{3}\left(2^j - 1\right) - \tfrac{1}{6}\left(\tfrac{1+(-1)^{j+1}}{2}\right),$$

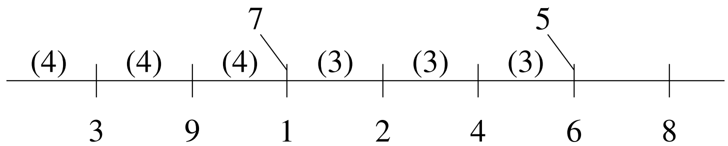
where $j = \left\lfloor \log\left(\frac{3n+2}{2}\right) \right\rfloor$. This form also has the advantage that it quickly gives an asymptotic ($n \to \infty$) evaluation for $U(n)$, namely

$$(6.16)\quad U(n) = jn - \tfrac{2^{j+2}}{3} + \tfrac{1}{2}\log(n) + O(1),$$

where $j = j(n)$ is defined above. The results of (6.15) and (6.16) are derived by Hadian in [10].

7. Remarks about the Table and the Trees

The trees below represent only a small sample of the trees constructed for the table in Section 5. Only the more involved trees with the most novel results are given. No tree was found that gives better (i.e., quieter) results than the modified entropy procedure R_{E^*}. However, there is reason to believe that higher-step entropy procedures may improve some of our results. This is based on the fact that in several situations that arise, the 2-step entropy is a clear improvement on the 1-step entropy; we give one illustration that arises under R_E for $n = 9$. After seven comparisons, one of the nodes of the tree has associated with it 21 possible states of nature, which we represent by the diagram:



The slanting lines indicate that 5 belongs somewhere below 6 and 7 belongs somewhere below 1. If we insert 5 first, it has six spaces in which to go, and the number of cases (or relative probability) for each is shown by the number in parentheses. The 1-step entropy procedure requires that we compare 5 with 1 to obtain the (12, 9) split rather than 5 vs. 9, which gives a (13, 8) split. However, the 2-step entropy procedure compares the four-way split (6, 6, 3, 6) (which has a unit of noise) for the former start with the four-way split (4, 4, 7, 6) for the latter start. The (4, 4, 7, 6) split is preferred under 2-step entropy since its 2-step reduction in entropy is

(7.1) $\frac{8}{21}\log\left(\frac{21}{4}\right) + \frac{7}{21}\log\left(\frac{21}{7}\right) + \frac{6}{21}\log\left(\frac{21}{6}\right) = 1.957\ldots$

compared to 1.952... for the (6, 6, 3, 6) split.

The procedure R_N represents an attempt to use our above results about noise in the construction of a procedure, and the results are quite good. In fact, the procedure R_N appears to be better than the 1-step entropy procedure R_E but not as good as the modified entropy procedure R_{E^*}.

The symbol S in our tree denotes a branch that is symmetrical to or equivalent to another branch to its left, which is further developed. The symbol H, with the integer j on the last arrow leading to it, means that the concluding steps starting from this point are obvious noiseless insertions that require an additional expected number $H(j)$ of comparisons (starting at that node and including it in the count). The symbol H_1 indicates that the remaining steps are not insertions, but they are still obvious and noiseless so that the same result (6.4) applies; we can regard the H's and H_1's as equivalent. The circled integers between the two forks of a noisy node are the number of noise units at that node.

It appears to be true that no noise can arise at a node that corresponds to a total of eight or fewer cases (i.e., states of nature), but this has not been proved.

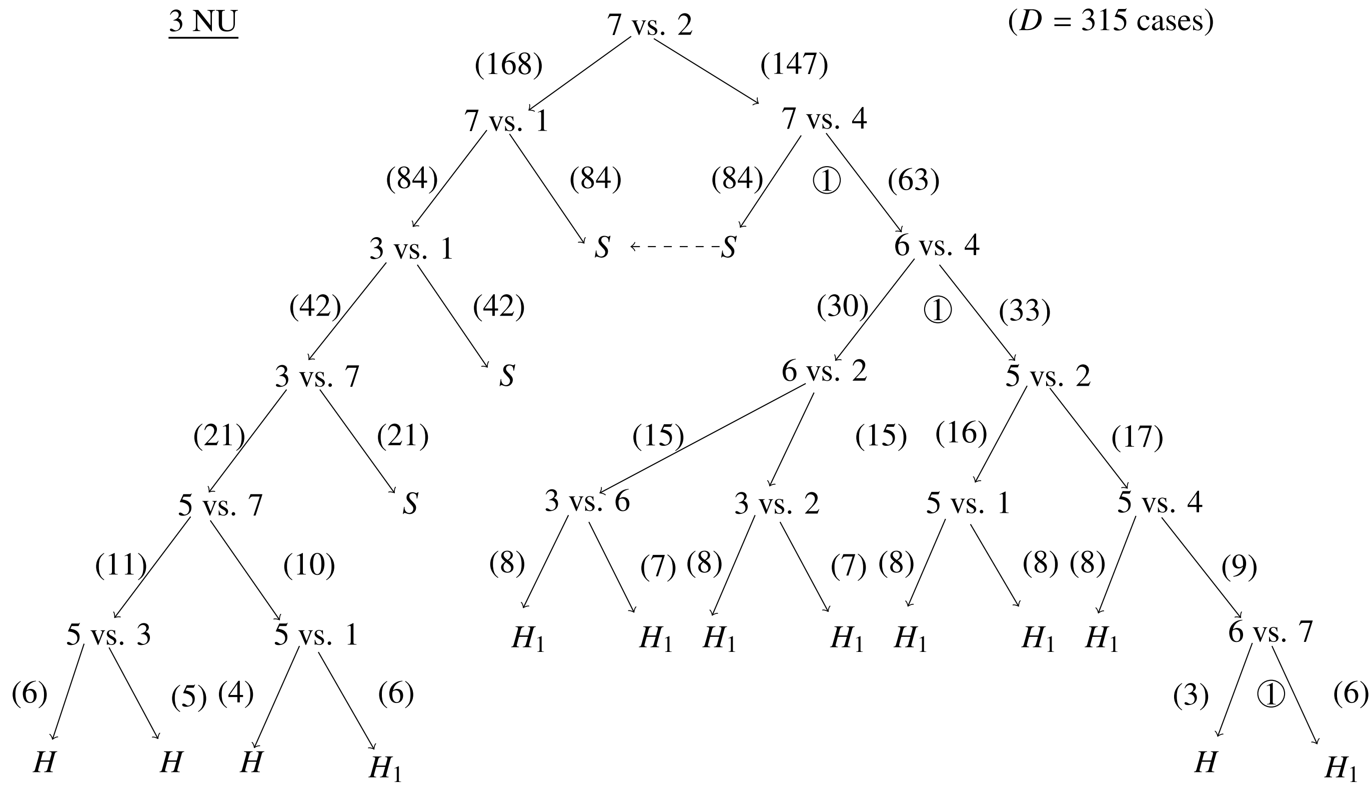
None of the procedures used contained any noisy nodes that were not simple.

Each of the trees below starts after the p pairings associated with complete pairing; here, p is the highest power of 2 that factors into $n!$. Hence, the total number of cases (or states of nature) at the top of the tree is $D = \frac{n!}{2^p}$, which is the common denominator in the table in Section 5.

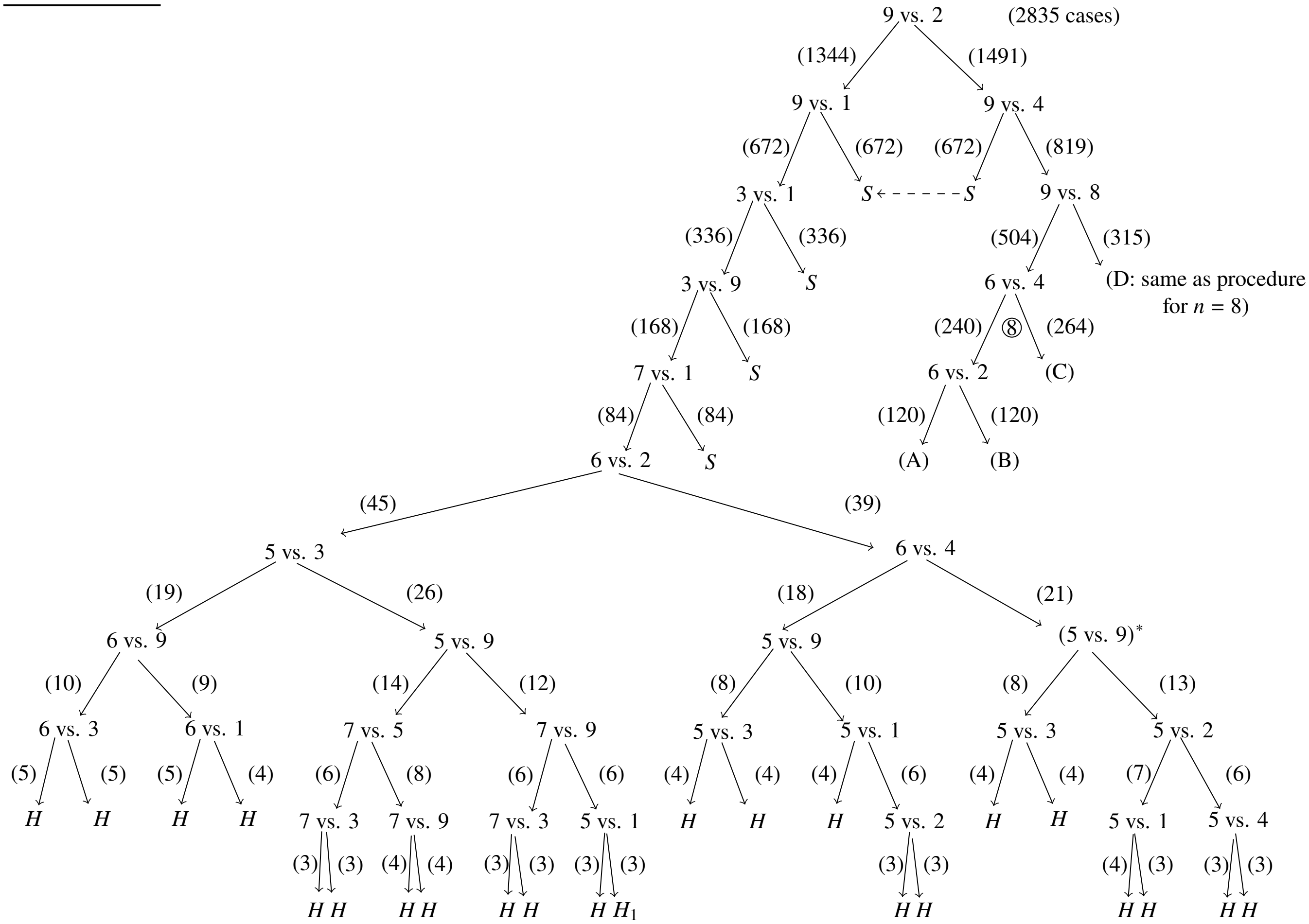
Since there are three noise units, the expectation for $n = 7$ under R_{E^*} is $4 + H(D) + \frac{3}{D} = 12\frac{121}{315} =$ 12.384... For $n = 8$, the procedure R_{E^*} is exactly the same except for three extra pairings (7 vs. 8, 6 vs. 8, and 4 vs. 8) at the outset. Hence, the expectation under R_{E^*} for $n = 8$ is $15\frac{121}{315} = 15.384\ldots$ It is conjectured that these are the best possible results for $n = 7$ and 8, but this has yet to be proved.

Cesari [4] has shown that no noiseless procedure exists for n = 7. With the aid of our results above, one could try to show that no procedure with NU < 3 exists, but this has not been attempted.

$\underline{R_{E^*} \text{ for } n = 7}$

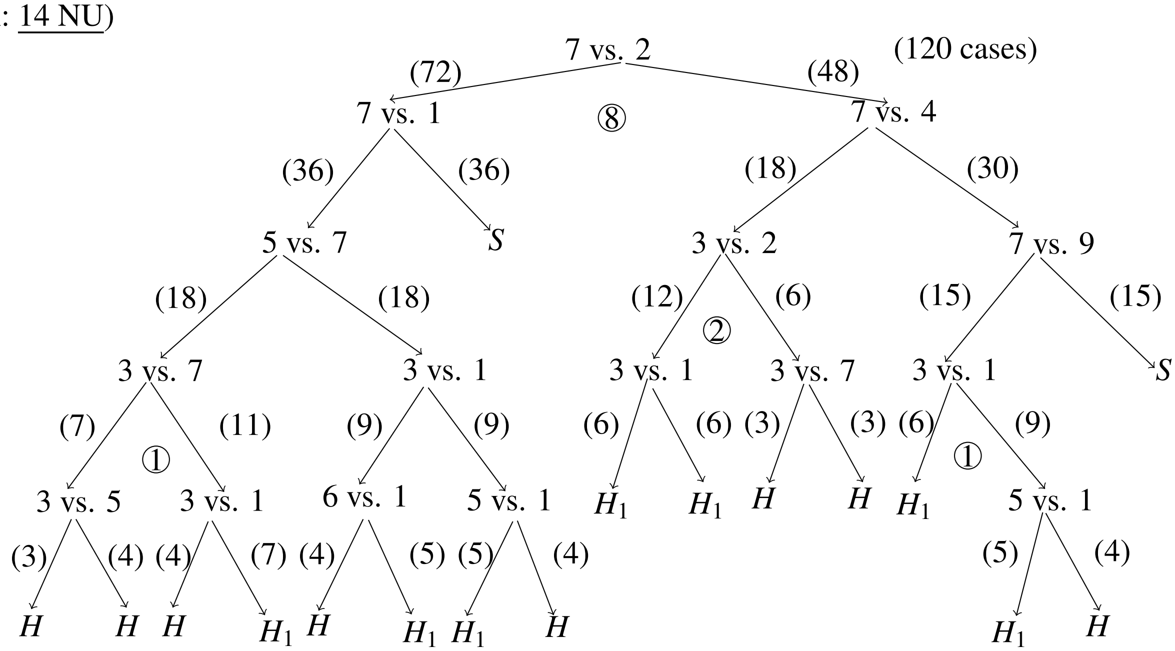


$\underline{R_N \text{ for } n = 9}$



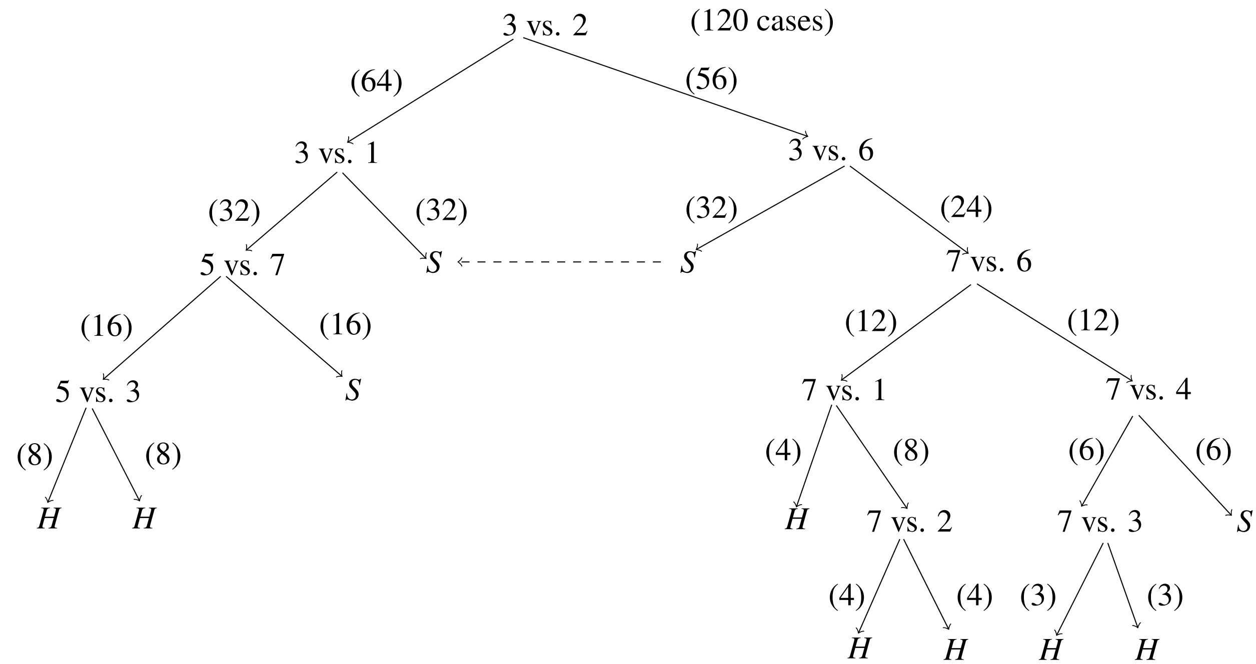
The total noise is 34 NU, and hence $E\{T|R_N\} = 18\frac{1574+34}{D} = 18\frac{1608}{2835} = 18\frac{536}{945} = 18.567\ldots$ Two-step entropy was used only at (*).

(A: $\underline{14 \text{ NU}}$)

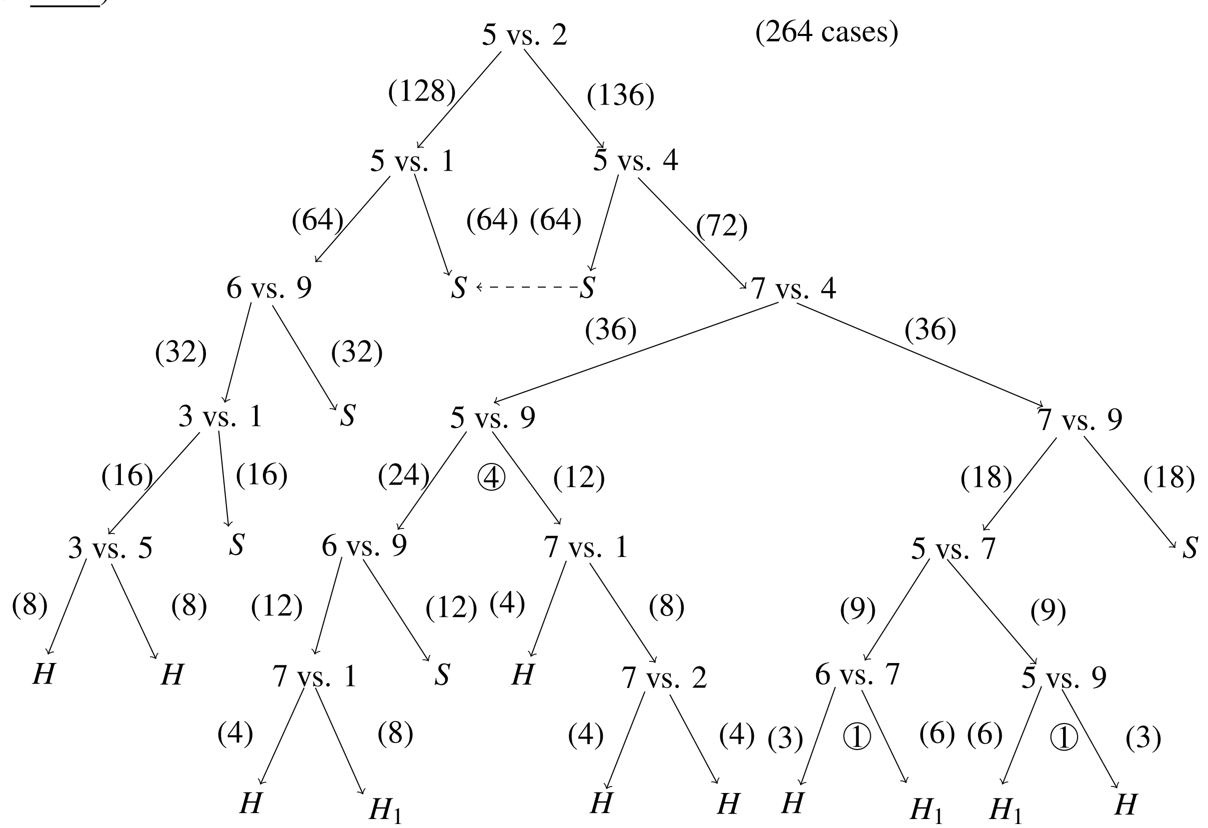


Continuation of R_N for $n = 9$

(B: No Noise)

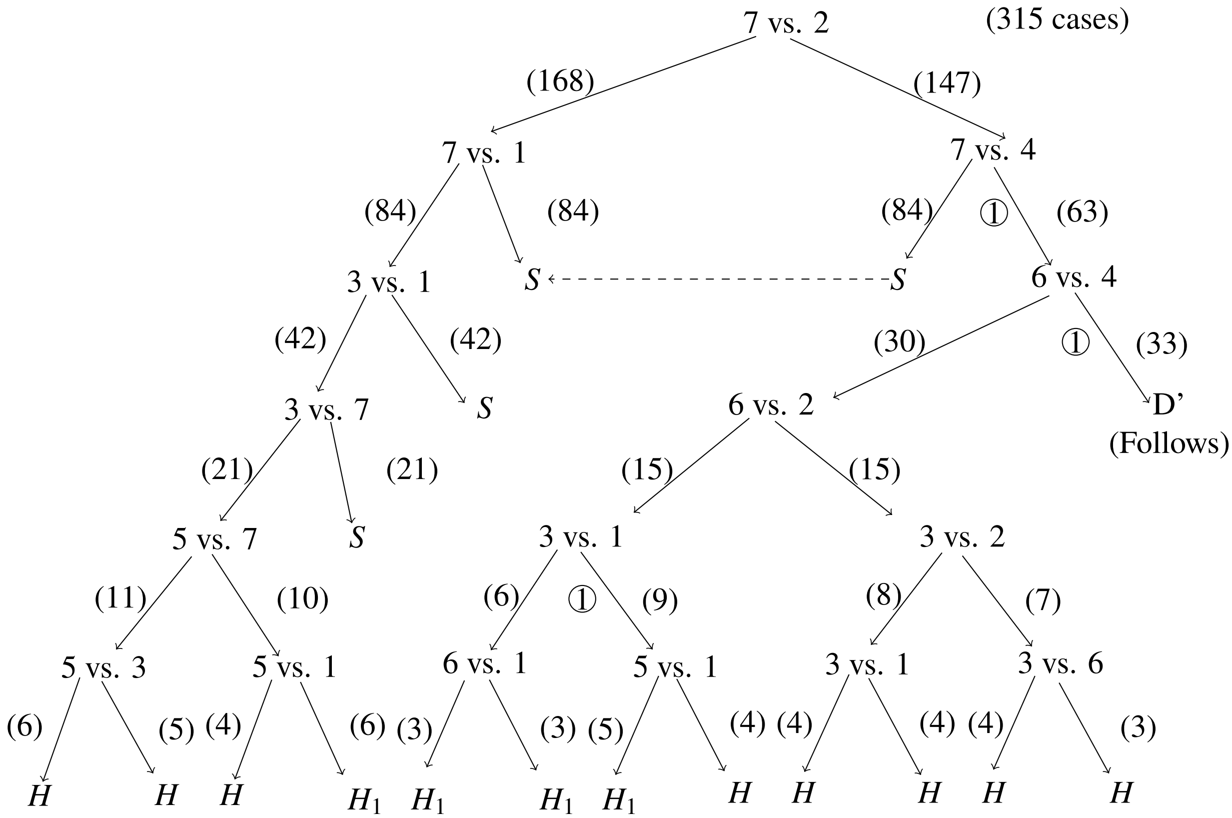


(C: 8 NU)

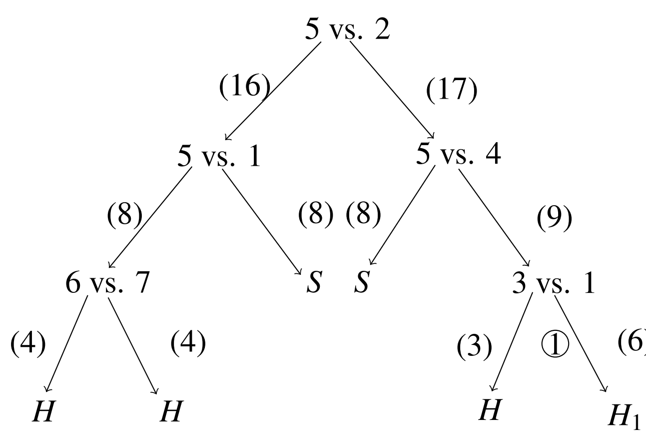


Continuation of R_N for $n = 9$

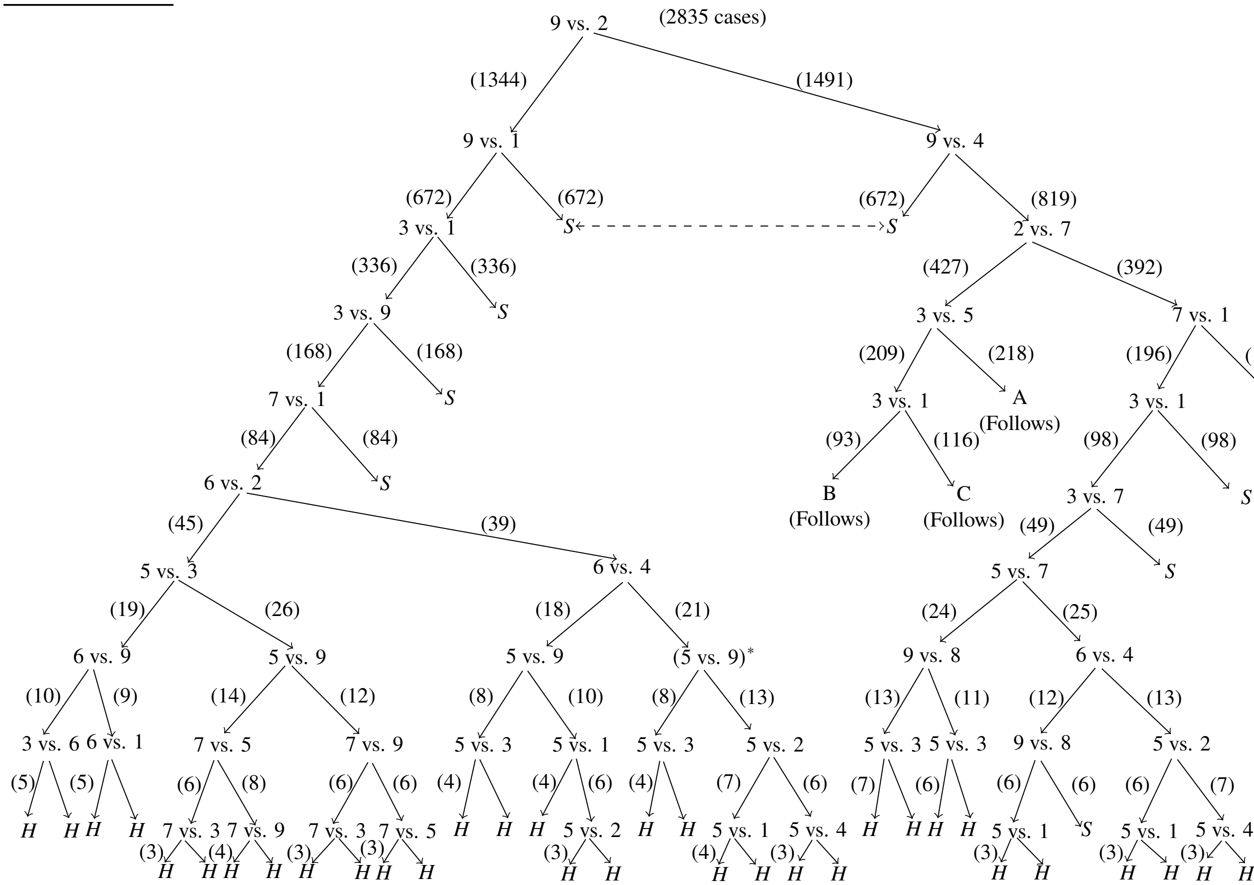
(D: 4 NU)



D':

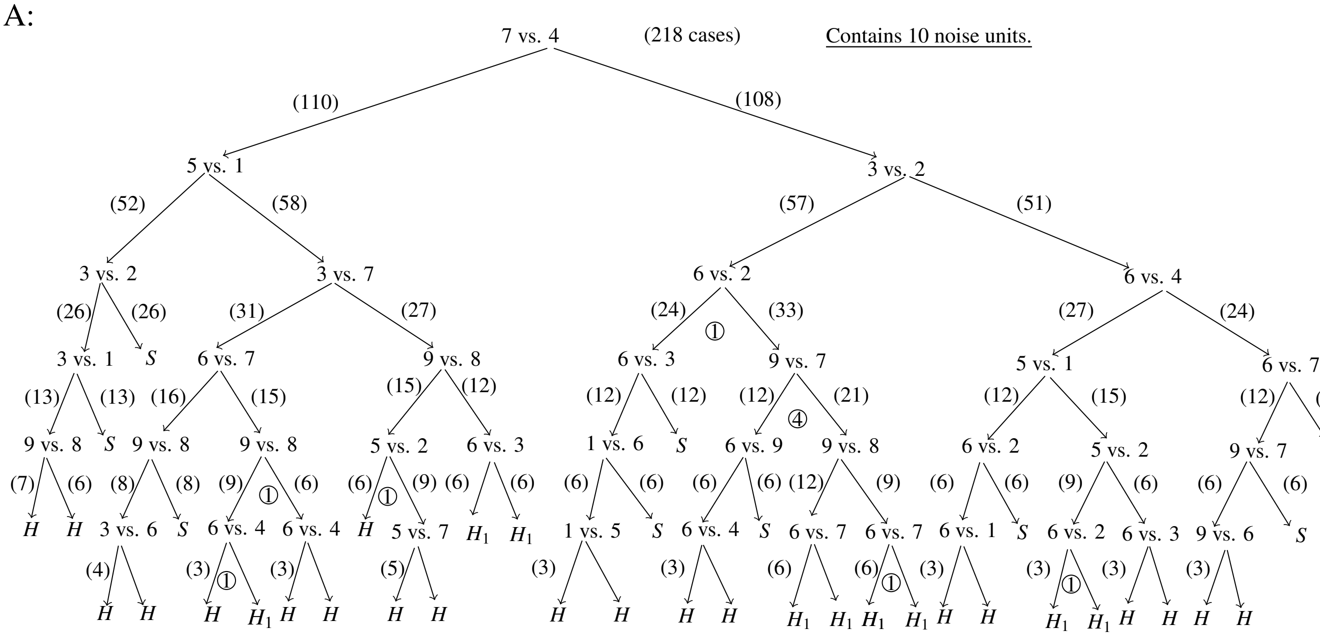


$\underline{R_{E^*} \text{ for } n = 9}$



Two-step entropy was used only at (*) above to avoid one unit of noise, which becomes 32 because of the multiplicities (S). The total remaining number of noise units (NU) is 18, and hence, $E\{T|R_{E^*}\} = 18\frac{1574+18}{D} = 18\frac{1592}{2835} = 18.562\ldots$ To get the result for R_E, we add $18 + 32 = 50$ NU, and the result is $18\frac{1624}{2835} = 18.573\ldots$

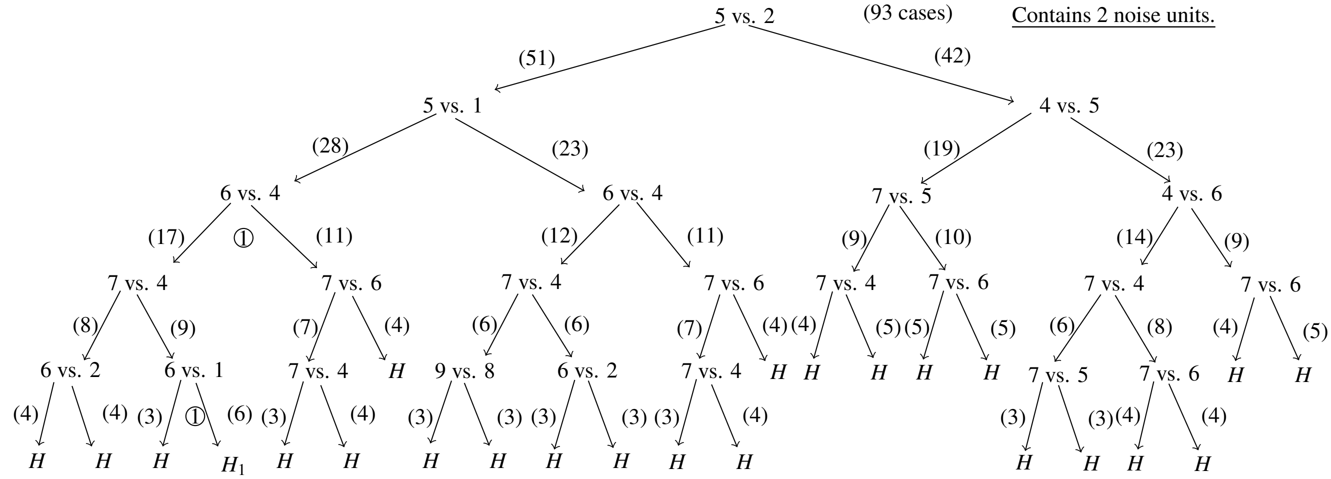
A:



Continuation of R_{E^*} for $n = 9$

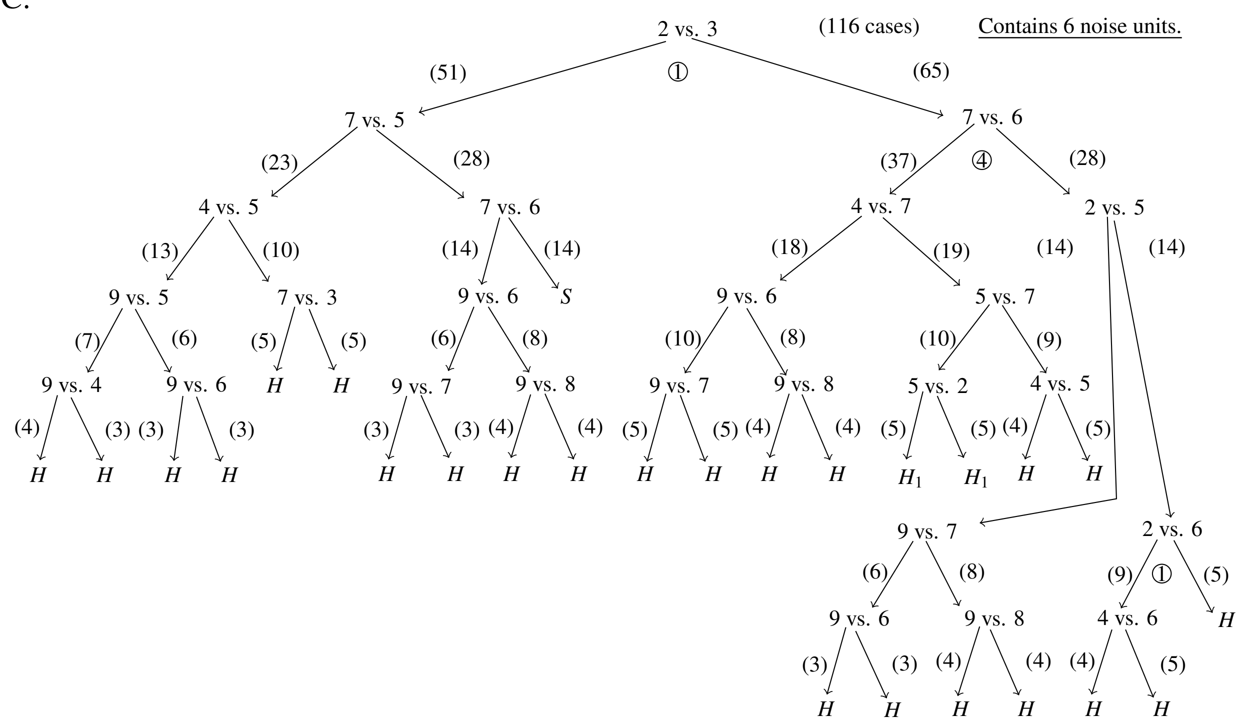
B:

(93 cases) Contains 2 noise units.



C:

(116 cases) Contains 6 noise units.



ACKNOWLEDGMENT

The author wishes to thank A. Hadian of the University of Minnesota for many stimulating conversations on this paper. In particular, it was he who recognized that the dynamic programming approach in (6.3) could be applied to the problem of ordering all n numbers.

REFERENCES

[1] R. C. Bose and R. J. Nelson. A sorting problem. *J. ACM*, 9(2):282–296, April 1962.

[2] W. H. Burge. Sorting, trees, and measures of order. *Information and Control*, 1(3):181–197, 1958.

[3] L. Carroll. *Lawn Tennis Tournaments: The True Method of Assigning Prizes with a Proof of the Fallacy of the Present Method*. Macmillan, 1883.

[4] Y. Cesari. *Questionnaire, codage et tris.* Problèmes de traitement de l'information. Centre Nat. de la Recharche Scient., Inst. Blaise Pascal, 1968.

[5] H. A. David. Tournaments and paired comparisons. *Biometrika*, 46(1-2):139–149, 06 1959.

[6] H. A. David. *The Method of Paired Comparisons*. Griffin's statistical monographs & courses. Hafner Publishing Company, 1963.

[7] F. Dubail. *Algorithmes de questionnaires optimaux au sens de différents critères*. Université de Lyon, France, 1967.

[8] L. R. Ford Jr. and S. M. Johnson. A tournament problem. *The American Mathematical Monthly*, 66(5):387–389, 1959.

[9] W. A. Glenn. A comparison of the effectiveness of tournaments. *Biometrika*, 47(3-4):253–262, 12 1960.

[10] A. Hadian. Optimality properties of various procedures for ranking n different numbers using only binary comparisons. Technical Report 117, Department of Statistics, University of Minnesota, Minneapolis, MN, USA, 1969.

[11] T. N. Hibbard. Some combinatorial properties of certain trees with applications to searching and sorting. *J. ACM*, 9(1):13–28, January 1962.

[12] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

[13] K. E. Iverson. A programming language. In *Proceedings of the May 1-3, 1962, Spring Joint Computer Conference*, AIEE-IRE '62 (Spring), page 345–351, New York, NY, USA, 1962. Association for Computing Machinery.

[14] S. S. Kislitsyn. On a bound for the smallest average number of pairwise comparisons necessary for a complete ordering of N objects with different weights. *Vestnik Leningrad University. Series 1: Mathematics, Mechanics, and Astronomy*, 18(1):162–163, 1962.

[15] S. S. Kislitsyn. An improved bound for the least mean number of comparisons needed for the complete ordering of a finite set. *Vestnik Leningrad University. Series 1: Mathematics, Mechanics, and Astronomy*, 19(4):143–145, 1963.

[16] R. J. Maurice. Selection of the population with the largest mean when comparisons can be made only in pairs. *Biometrika*, 45(3-4):581–586, 12 1958.

[17] C. F. Picard. *Théorie des questionnaires*. Number 20 in Les Grands problèmes des sciences. Gauthier-Villars, 1965.

[18] M. Sandelius. On an optimal search procedure. *The American Mathematical Monthly*, 68(2): 133–134, 1961.

[19] J. Schreier. O systemach eliminacji w turniejach. *Mathesis Polska*, 7:154–160, 1932.

[20] J. Słupecki. On the systems of tournaments. *Colloquium Mathematicae*, 2(3-4):286–290, 1951.

[21] M. Sobel. Optimal group testing. Proc. Colloq. Inform. Theory, Debrecen 1967, 411–488, 1968.

[22] M. Sobel and P. A. Groll. Group testing to eliminate efficiently all defective in a binomial sample. *The Bell System Technical Journal*, 38(5):1179–1252, 1959.

[23] H. Steinhaus. Mathematical snapshots. New York: Oxford University Press, 1950.

[24] H. Steinhaus. One hundred problems in elementary mathematics. Popular Lectures in Mathematics. Oxford etc.: Pergamon Press; Warszawa: PWN-Polish Scientific Publishers, 1963.

[25] H. Steinhaus. Some remarks about tournaments. Calcutta Mathematical Society, *Golden Jubilee Commemoration Volume (1958-1959)*, Part 2, 323–327. Golden Jubilee Celebration Committee, 1959.